

Signal Processing for Improved MPEG-based Communication Systems

Signal Processing for Improved MPEG-based Communication Systems

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de Technische Universiteit
Eindhoven, op gezag van de rector magnificus prof.dr.ir. F.P.T. Baaijens,
voor een commissie aangewezen door het College voor Promoties, in het
openbaar te verdedigen op woensdag 9 december 2015 om 16:00 uur

door

Onno Eerenberg

geboren te Zwolle

Dit proefschrift is goedgekeurd door de promotor en de samenstelling van de promotiecommissie is als volgt:

voorzitter:	prof.dr.ir. J.W.M. Bergmans
1 ^e promotor:	prof.dr.ir. P.H.N. de With
copromotor(en):	prof.dr. R.M. Aarts
leden:	prof.dr. C. Hentschel (Brandenb. Univ. of Technol. Cottbus, Germany) prof.dr. S. Sherratt (Univ. of Reading, United Kingdom) prof.dr.ir. J.J. Lukkien
adviseur(s):	dr. P. Hofman (Philips IP&S)

Het onderzoek of ontwerp dat in dit proefschrift wordt beschreven is uitgevoerd in overeenstemming met de TU/e Gedragscode Wetenschapsbeoefening.

Nulla tenaci in via est –
Voor de volhouder is geen weg onbegaanbaar
(Spyker Automobielen N.V.).

CIP-DATA LIBRARY TECHNISCHE UNIVERSITEIT EINDHOVEN

Onno Eerenberg

Signal Processing for Improved MPEG-based Communication Systems / by Onno Eerenberg. - Eindhoven : Technische Universiteit Eindhoven, 2015.

A catalogue record is available from the Eindhoven University of Technology Library

ISBN: 978-90-386-3979-6

NUR: 959

Trefw: videocompressie / personal video recording / MPEG-2 / H.264/MPEG4-AVC / trick play / DVB-H / video coding artifact detection.

Subject headings: conventional video navigation / advanced video navigation / MPEG-2-compliant video navigation / DVB-H link layer / mosquito and ringing artifact location detection.

Cover design: D.J.M. Frishert.

Printed by: Dereumaux.

Copyright © 2015 by O. Eerenberg

All rights reserved. No part of this material may be reproduced or transmitted in any form or by any means, electronic, mechanical, including photocopying, recording or by any information storage and retrieval system, without the prior permission of the copyright owner.

Summary

Signal Processing for Improved MPEG-based Communication Systems

This thesis describes improvements for MPEG-based consumer communication systems and focuses on three areas. The first area addresses intra-program video navigation for disk-based storage media, where three forms of video navigation are investigated enabling conventional as well as more advanced forms of video navigation. The second area presents an efficient and robust data link layer for DVB-H, a standard targeting battery-powered mobile television reception. The improved link layer results in a higher robustness and efficiency. The third area addresses picture quality for digital-television reception, presenting two detection systems for locating visual-coding artifact regions, which are potentially contaminated with either mosquito noise or ringing. The location information is used to attenuate the detected coding noise. The emphasis of the presented work is on embedded system solutions to be integrated into existing consumer platforms. The three areas are briefly summarized below.

In this thesis, three navigation techniques are presented for disk-based storage systems. The first navigation technique equals that of full-frame fast-search and slow-motion playback and is suitable for a push-based architecture, enabling deployment in a networked client-server system setup. Networked full-frame fast-search video navigation is based on re-using intra-coded MPEG-compressed normal-play video pictures. The proposed solution divides the signal processing for navigation over both recording and navigation playback operation mode. It is based on the finding of characteristic point information during recording, revealing the storage locations of intra-coded pictures, which are then re-used for generation of a fast-search navigation sequence.

Furthermore, in order to adapt the frame rate, refresh-rate, bit rate and playback speed, the solution employs repetition pictures, which repeat normal-play reference pictures. On the basis of field-based repetition pictures, rendering control at field level is obtained, enabling efficient removal of field-based video information (interlace kill), thereby avoiding motion judder during nav-

igation, enabling the navigation method to be applied to both progressive and interlaced video formats. Slow-motion video navigation is implemented in a similar fashion. When applying repetition pictures to control the Quality of Experience (QoE) during navigation, the refresh-rate should not drop below $1/3$ of the frame rate, otherwise a slide-show effect occurs, whereby the viewer loses the fast-search navigation experience. For a typical video broadcast at SD resolution and 4 Mbit/s, the required DSP processor load during recording requires a cycle frequency of 5 MHz, while a cycle frequency of 22 MHz is required for full-frame fast-search video playback with speed-up factor 12 and 5 MHz for slow-motion with playback speed 0.5. Both fast-search and slow-motion video navigation can be operated with a reduced refresh-rate and thus a considerably lower cycle frequency.

Based on drawbacks associated to full-resolution fast-search trick play, a video navigation technique is presented based on a hierarchical mosaic representation. This screen is composed of re-used MPEG-compressed subpictures, avoiding transcoding of pictorial data. Hierarchical mosaic screens enable instant overview of video information associated with a large temporal interval, thereby eliminating the individual picture refresh-rate from the final video navigation rendering. Each subpicture is coded at a fixed bit cost, thereby simplifying the construction of the final mosaic screen in the compressed domain. A fixed-cost subpicture is achieved by dividing each subpicture into a set of “*mini slices*”, which are also encoded at a fixed bit cost. Furthermore, when subpictures use P-type coding syntax, new mosaic screens can be constructed using predictive coding, based on re-used subpictures available at the MPEG decoder. Both aspects clearly reduce the complexity of the implementation. The continuous derivation of subpictures for mosaic screens requires only a low fraction of the computation complexity, because this intra-coded normal-play pictures appear at a rate of only 2 Hz, which results in a processing load of 0.3 Hz, when a scene duration of 3 seconds is used. It was found that this system can be implemented with the same architecture as the first navigation solution, because the processing required for the construction of a mosaic screen has a high resemblance with the fast-search full-frame navigation solution. We therefore expect that the involved playback processing for mosaic-screen navigation will show a similar throughput and DSP cycle load.

Finally, an audio-enhanced dual-window video navigation technique is presented, combining normal-play audiovisual fragments, with a down-scaled fast-search information signal. This representation addresses human perception which employs both visual and auditory queues. Hereby detailed normal-play information is rendered in a main window, while a coarse overview is provided by fast-search information rendered in a second picture-in-Picture (PiP) window. Due to simultaneous rendering, a viewer can switch between the two information signals, perceiving either a fast- or a detailed overview, guiding the viewer to the requested audiovisual fragment.

The main navigation signal is based on re-using normal-play audio and video fragments of typical length of 3 seconds (approx. 6 GOPs), which are fully decoded. The fast-search navigation signal is derived from decoded normal-play intra-coded pictures, which are scalable MPEG-2 decoded already during recording. The scalability enables downscaling while decoding, since the picture quality is lower. These lower-quality pictures are stored, next to the other essential Characteristic Point Information (CPI) in the metadata database. During playback, the concatenated stream fragments are modified to ensure MPEG-2-compliant formatting and enabling seamless decoding. This modification involves amongst others audio padding, removal of predictive images without reference and modification of the time base for navigation. Scalability is implemented by partial decoding of the intra-compressed data, which is worked out for both MPEG-2 and H.264/MPEG4-AVC standards. The obtained picture quality for both standards is in the range of 23-32 dB, which is lower than the normal-play quality but sufficient for navigation purposes.

The three proposed navigation solutions show a high commonality with respect to the required signal processing and the execution architectures, so that they can be jointly implemented in one overall architecture.

The second contribution of this thesis aims at improving the DVB-H link layer for mobile battery-powered MPEG communication. The problem of the standard solution in a receiver is that it does not provide sufficient robustness and is also based on inefficient bandwidth usage. The solution is based on locally obtained reliability and location information, in the form of 2-bit erasure flags, Internet Protocol Entry Table (IPET) and Correct Row Index Table (CRIT). The reliability information is derived on the basis of dual-stage FEC decoding, while the location information is derived from correctly received broadcast data. Hereby, the primary FEC is performed by the channel decoder, while the secondary FEC is integrated in the DVB-H link layer. The usage of the reliability information derived from the primary FEC is employed in two ways. First, this reliability information is applied for error and erasure decoding by the secondary FEC stage. For the situation that after this second FEC stage an MPE-FEC frame is still incorrect, this reliability information is used for a second time, in combination with reliability information derived from the secondary FEC decoding stage, supplemented with the IPET and CRIT information. In this way, correctly received and corrected IP datagrams are extracted from the defect MPE-FEC frame. Using this new link layer concept, a significantly improved performance was realized. As a result, the robustness for retrieving completely correct MPE-FEC frames improves with approximately 50%. However, the performance curves cluster around the same critical performance degradation point, resulting in a minor graceful signal degradation when errors cannot be avoided. Efficiency is achieved by avoiding IP datagram duplication, which means that correct IP datagrams, either correctly received or corrected by FEC

are only forwarded once to the IP-stack, optimizing on bandwidth and contributes to a reduced power consumption.

MPEG-based video broadcasting introduces coding artifacts due to the quantization of signal components when the system is operated at limited bandwidth. Reduction of these coding artifacts requires a detection stage followed by an attenuation stage, whereby the detection should address different types of coding noise. Two block-based artifact-location detection systems are presented, either operating in the spatial domain or frequency domain, for detecting the locations of mosquito noise and ringing noise patterns. The two detectors generate a filter-strength control signal, which is used by an entropy-based adaptive low-pass filter, attenuating the local contamination while preserving the overall picture quality. The activity of the detected noise or ringing is classified in three simple indications, like “flat and/or low-frequency”, “noise contaminated (mosquito noise and ringing)” and “texture”. The usage of these signal features is employed in two ways. First, on the basis of the block-based signal classification a simplified video model is derived, using the surrounding blocks of the considered area and classify those blocks in a similar way. These surrounding blocks act as additional context information suitable for context reasoning about the visibility of possible coding noise. This gives a more robust location detection of potentially noise-pattern contaminated regions. Second, these block-based signal features are also employed afterwards to expand the detection signal employing a diamond-shaped aperture. In this way, the detection signal becomes more consistent as a filter control signal and increases the area coverage of the detected contaminated region. Artifact detection in the spatial domain, employing a detection kernel of 11×7 pixels based on a fixed block size of 3×3 pixels has a detection score of 80 and 86 %, for JPEG-compressed pictures with $Q = 25, Q = 50$, respectively. For the frequency-domain detection system, employing a detection kernel of 20×12 pixels based on a fixed block size of 4×4 pixels, this detection score is 98 and 99 %, for JPEG-compressed pictures with $Q = 50, Q = 25$, respectively. The spatial-domain detection method appeared to be slightly better and more accurate with the respect to the object contours. The spatial-domain solution provides a moderate average increase in PSNR of +0.05 dB, but with locally strongly improved areas. Furthermore, the combined system avoids excessive image blur from filtering, e.g noticeable from the positive PSNR contributions, which is due to the context reasoning in the kernel area and the detailed decision making.

Samenvatting

Signaalbewerking voor verbeteringen in mpeg-gebaseerde communicatiesystemen

Dit proefschrift beschrijft verbeteringen in drie toepassingsgebieden van mpeg-gebaseerde videocommunicatiesystemen. De eerste toepassing gaat over videonavigatie binnen een tv-programma voor diskgebaseerde consumentenproducten, waarvoor drie vormen van videonavigatie zijn onderzocht, die geschikt zijn voor zowel conventionele als geavanceerde vormen van videonavigatie. Het tweede toepassingsgebied presenteert een efficiënte en robuuste DVB-H data link layer voor een mobiele digitale tv-ontvanger met batterijvoeding. De derde toepassing gaat over beeldkwaliteitsverbetering van digitale tv-ontvangers. Er worden twee detectiesystemen gepresenteerd voor de detectie van potentiële zichtbare mosquito-ruis en ringing-effecten tengevolge van videocompressie. De verkregen locatie-informatie wordt toegepast voor het reduceren van de gedetecteerde eerder genoemde coderingsruis. De drie toepassingsgebieden zijn hieronder samengevat.

In dit proefschrift worden drie navigatietechnieken gepresenteerd voor diskgebaseerde opslagsystemen. De eerste techniek omvat de versnelde en vertraagde videoweergave op basis van volledige tv-beelden. Deze methode is geschikt voor een push-gebaseerde systeemarchitectuur, die kan worden gebruikt in een genetwerkt client-serversysteem. Versnelde navigatie met volledige beelden in een genetwerkte systeemopzet is gebaseerd op het hergebruik van intraframe-gecodeerde mpeg tv-beelden. De gekozen oplossing verdeelt de signaalbewerking over de opname- en weergaveperiode. De oplossing is gebaseerd op het bepalen van de karakteristieke informatie tijdens opname die de opslaglocaties aangeeft van intraframe-gecodeerde tv-beelden, die worden hergebruikt voor het genereren van de versnelde navigatievideo.

Voor het aanpassen van de beeldfrequentie, beeldverversingssnelheid (refresh-rate), bitsnelheid (bit rate) en weergavesnelheid, worden in de oplossing zo-

genaamde herhalingsbeelden toegepast, die zorgen voor het herhaald presenteren van een referentiebeeld. Met field-gebaseerde herhalingsbeelden kan de weergave van geïnterlineerde videobeelden worden gecontroleerd. Hiermee wordt het mogelijk om field-gebaseerde videoinformatie te verwijderen (interlace kill), waarmee bewegingstrilling tijdens de navigatie wordt vermeden, zodat deze navigatiemethode geschikt is voor progressive en geïnterlineerde videoformaten. De vertraagde weergave wordt op een vergelijkbare manier gerealiseerd. Wanneer herhalingsbeelden tijdens versnelde videonavigatie worden toegepast voor het doorvoeren van Quality-of-Service (QoS), dan moet de beeldverversing (refresh-rate) niet beneden $1/3$ van de beeldfrequentie zakken, anders ontstaat er een diashow effect, waardoor de kijker het gevoel van versneld afspelen verliest. Voor een typische tv-uitzending met standaardresolutie (SD) op 4 Mbit/s resulteert de signaalbewerking tijdens opname in een 5 MHz kloksnelheid van de DSP-processor, terwijl een DSP-klokfrequentie van 22 MHz nodig is voor twaalfvoudig versnelde weergave met volledige beelden, en met 5 MHz kloksnelheid voor de DSP-processor voor vertraagd afspelen op halve snelheid. Zowel versnelde als vertraagde navigatieweergave kan worden gerealiseerd met een gereduceerde refresh-rate en dus met een aanzienlijk lagere DSP-klokfrequentie.

De tweede gepresenteerde navigatiemethode is gebaseerd op een hiërarchisch mozaïekschermbestuur, die probeert de nadelen van versnelde weergave met volledige beelden te vermijden. Het mozaïekschermbestuur bestaat uit hergebruikte gecomprimeerde deelbeelden, waardoor opnieuw coderen niet meer nodig is voor de constructie van een mozaïekschermbestuur. Hiërarchische mozaïekschermbesturen bieden een instantaan overzicht van videoinformatie over een groot tijdsinterval, dat onafhankelijk is van de weergavetijd van individuele beelden in de uiteindelijke videonavigatie. Ieder deelbeeld is gecomprimeerd met een constante hoeveelheid bits (bit cost), waardoor de constructie van een mozaïekschermbestuur met gecomprimeerde data wordt vereenvoudigd. Een deelbeeld met vaste bit cost is gerealiseerd door ieder deelbeeld op te splitsen in een set van "mini-slices" (series van vierkante blokken met beeldinformatie), die zelf ook weer op een vaste bit cost zijn gecodeerd. Wanneer de deelbeelden gebruik maken van de P-type coderingssyntax, dan kunnen nieuwe mozaïekschermbesturen worden geconstrueerd met behulp van predictive codering, door gebruik te maken van deelbeelden die al beschikbaar zijn bij de mpeg-decoder. Beide aspecten verminderen de complexiteit van de implementatie aanzienlijk. Het voortdurend afleiden van deelbeelden voor mozaïekschermbesturen vormt maar een klein gedeelte van de rekenkundige complexiteit, omdat deze intra-gecodeerde beelden verschijnen met een beeldfrequentie van 2 Hz, wat resulteert in een beeldverwerkingssnelheid van 0.3 Hz bij een scèneduur van 3 seconden. Uit onderzoek blijkt dat dit systeem kan worden geïmplementeerd met dezelfde architectuur als de eerste navigatiemethode, omdat de vereiste bewerkingen voor de constructie van een mozaïekschermbestuur gebaseerde videonavigatie een grote over-

eenkomst heeft met de versnelde navigatie met volledige beelden. Het is daarom aannemelijk dat de signaalbewerking voor mozaïekschermnavigatie een vergelijkbare bandbreedte en DSP-kloksnelheid nodig heeft.

Als laatste wordt een duovenster-videonavigatie gepresenteerd die met audio is uitgebreid, waarbij audiovisuele fragmenten worden gecombineerd met een versnelde zoekweergave, die wordt gemaakt van beelden met verlaagde resolutie. Deze navigatievorm appelleert aan zowel de visuele als de auditieve perceptie van de gebruiker. Bij deze combinatie worden gedetailleerde visuele programmafragmenten afgebeeld in een hoofdscherm en wordt de minder gedetailleerde maar versnelde navigatie afgebeeld in een tweede, kleiner venster als een Picture-in-Picture (PiP). Door het gelijktijdig presenteren van beide deelnavigaties kan de kijker dynamisch kiezen tussen een gedetailleerd en een minder gedetailleerd informatiesignaal, die hem naar de gewenste videofragmenten leiden.

Het gedetailleerde navigatiesignaal in het hoofdscherm is gebaseerd op hergebruikte audiovisuele fragmenten met een typische tijdsduur van 3 seconden (ongeveer 6 GOPS), die volledig gedecodeerd worden. Het versnelde navigatiesignaal is afgeleid van intraframe-gecomprimeerde beelden, welke tijdens opname zijn verkregen via schaalbare mpeg-decodering. Omdat deze navigatiebeelden een lagere kwaliteit hebben, kan een vereenvoudiging al worden gerealiseerd tijdens het decoderen door middel van schaalbaarheid. De benodigde beelden worden opgeslagen samen met andere essentiële karakteristieke informatie (Characteristic Point Information (CPI)) in de metadata database. Voor het vloeiend decoderen worden tijdens het afspelen de gecombineerde audiovisuele fragmenten aangepast voor het verkrijgen van een mpeg-standaardsignaal. Deze modificatie omvat o.a. het aanpassen van audio padding, verwijdering van predictive beelden zonder tijdsreferentie en aanpassing van de navigatietijdbasis. Schaalbaarheid is gerealiseerd door partiële decodering van intra-gecodeerde beelden en uitgewerkt voor zowel de mpeg-2 als de h.264/mpeg-4-standaard. De resulterende navigatiebeeldkwaliteit voor beide standaarden is in de orde van 23-32 dB en is lager dan die van de originele opname, maar is van voldoende kwaliteit voor navigatiedoeleinden.

De drie besproken navigatiemethoden vertonen een hoge mate van gelijkheid met betrekking tot de vereiste signaalbewerking and executiearchitectuur, zodat deze kunnen worden geïmplementeerd met dezelfde gemeenschappelijke architectuur.

De tweede bijdrage van dit proefschrift is gericht op het verbeteren van de DVB-H link layer voor mobiele digitale tv-ontvangers met batterijvoeding. Het probleem van de standaardimplementatie van het systeem is gebrek aan voldoende robuustheid en inefficiënt bandbreedtegebruik. De voorgestelde oplossing is gebaseerd op lokaal verkregen betrouwbaarheids- en locatie-informatie, in de vorm van 2-bit foutenvlaggen (erasure flags) en twee nieuwe, intern ge-

bruikte tabellen, namelijk de Internet Protocol Entry Table (IPET) en de Correct Row Index Table (CRIT). De betrouwbaarheidsinformatie is verkregen op basis van twee foutencorrigerende decoders (FEC), terwijl de locatie-informatie is afgeleid van de foutloos ontvangen data. Hierbij wordt de primaire FEC uitgevoerd door de kanaaldecoder, terwijl de secundaire FEC wordt berekend door de DVB-H link layer. Het gebruik van betrouwbaarheidsinformatie die is afgeleid van de primaire FEC wordt op twee manieren toegepast. Ten eerste wordt de betrouwbaarheidsinformatie gebruikt door de secundaire FEC voor het decoderen volgens de foutencorrectie-met-erasuremethode. Voor de situatie dat na de secundaire FEC een MPE-FEC-dataveld nog steeds fouten bevat, wordt de betrouwbaarheidsinformatie een tweede keer toegepast, dit in combinatie met betrouwbaarheidsinformatie afgeleid van de secundaire FEC decodering, die is aangevuld met de IPET- en CRIT-tabelinformatie. Op deze wijze worden correcte en gecorrigeerde IP datagrammen geëxtraheerd uit een defect MPE-FEC-dataveld. Gebruik van dit nieuwe link-layer-concept levert een significante verbetering op, waarbij de robuustheid voor het construeren van volledig correcte MPE-FEC-datavelden met ongeveer 50% wordt verbeterd. Helaas clusteren de prestatiegrafieken rond hetzelfde kritische degradatiepunt, hetgeen leidt tot een matige geleidelijke degradatie wanneer toch fouten optreden. De systeemefficiëntie is verhoogd door duplicatie van een IP-datagram te vermijden, zodat correcte IP-datagrammen maar één keer aan de netwerklag worden aangeboden. Hierdoor wordt de bandbreedte geoptimaliseerd wat bijdraagt aan een gereduceerd vermogensverbruik.

Tv-uitzendingen die gebaseerd zijn op de mpeg-standaard introduceren coderingsruis en artefacten zoals "ringing" tengevolge van het kwantiseren van signaalcomponenten, wanneer het tv-systeem met beperkte bandbreedte wordt gebruikt. Reductie van deze coderingsartefacten vereist een detectiestap gevolgd door een verzwakkingsstap in de signaalverwerking, waarbij de detectie verschillende soorten coderingsruis moet identificeren. Er worden twee detectiesystemen geïntroduceerd, die of in het spatiële domein of in het frequentiedomein werken voor het bepalen van de locaties voor het optreden van mosquito-ruis en ringing. Beide detectiesystemen produceren een controle-sigitaal, dat door een entropiegebaseerd adaptief laagdoorlaatfilter wordt gebruikt voor het verzwakken van de lokale coderingsruis met behoud van de globale beeldkwaliteit. De activiteit van de gedetecteerde ruis of ringing wordt geclassificeerd in drie eenvoudige indicaties, zoals "vlak en/of laagfrequent", "ruisvervuild" (mosquito-ruis en ringing) en "textuur". Het gebruik van deze signaalclassificatie wordt op twee manieren toegepast. Ten eerste wordt een eenvoudig videomodel afgeleid voor alle videoblokken in het omliggende gebied op basis van dezelfde classificatie voor elk blok. Deze naburige blokken leveren additionele contextinformatie, die geschikt is voor het redeneren over contextuele informatie en de mogelijke zichtbaarheid van de coderingsruis. Dit

levert een meer robuuste locatiedetectie op van beeldgebieden die mogelijk ruispatronen bevatten. Ten tweede worden de blokgebaseerde signaaleigenschappen tevens toegepast na de ruisdetectie voor het expanderen van het detectiesignaal op basis van een diamantvormige filterapertuur. Op deze manier ontstaat er een meer consistent filtercontrolesignaal en neemt de grootte van het te analyseren contextuele gebied toe. Artefactdetectie in het spatiële domein, gebruikmakend van een detectiegebied (kernel) van 11×7 pixels en een vaste blok grootte van 3×3 pixels heeft een detectiescore van respectievelijk 80 en 86 %, voor jpeg-gecomprimeerde beelden met $Q = 25$, $Q = 50$. Voor detectie in het frequentiedomein op basis van een detectiekern met 20×12 pixels gebaseerd op een vaste blok grootte of 4×4 pixels, is deze detectiescore respectievelijk 98 en 99 % voor jpeg-gecodeerde beelden met $Q = 50$, $Q = 25$. De detectiemethode in het spatiële domein is iets beter en meer accuraat met betrekking tot de objectcontouren. De spatiële oplossing levert een redelijk gemiddelde verbetering in PSNR van +0.05 dB, maar met lokaal sterk kwaliteitsverbeterde gebieden. Daarnaast voorkomt deze oplossing lokale beeldonscherpte (blur) door een te sterke filtering, die wordt geïllustreerd door de positieve bijdrage aan de PSNR. Deze verbetering wordt veroorzaakt door de contextinformatie over de lokale beeldomgeving en de gedetailleerde besluitvorming over de coderingsruis.

Contents

Summary	i
Samenvatting	v
Contents	xi
1 Introduction	1
1.1 Preliminaries	1
1.2 Background	3
1.3 Research scope and problem description	6
1.3.1 Video Navigation for Digital Recording	6
1.3.2 Efficient and Robust DVB-H Link Layer	7
1.3.3 Block-based Visual Artifact-Location Detection	8
1.4 Contributions of the research	8
1.5 Outline and scientific background	10
2 Technology overview	13
2.1 Introduction	13
2.2 MPEG-2 Standard	15
2.2.1 MPEG-2 Part 2: Video	15
2.2.2 MPEG-2 Part 1: Systems	19
2.2.3 MPEG-2 Part 3: Audio	24
2.3 H.264/MPEG4-Advanced Video Coding	26
2.3.1 Intra-Macroblock Video Compression	27
2.3.2 Inter-Macroblock Video Compression	28
2.4 Video Coding Artifacts	29
2.5 Personal Video Recording	31
2.5.1 Intra-program video navigation	33
3 MPEG-2-compliant video navigation	41
3.1 Introduction	41

3.2	Proposed video navigation use cases	43
3.3	Conceptual MPEG-2-compliant video navigation	45
3.4	Networked full-frame video navigation	50
3.4.1	Usage of repetition pictures	50
3.4.2	System aspects for video navigation algorithms	54
3.4.3	Algorithm for MPEG-2-compliant fast-search trick play	60
3.4.4	Algorithm for MPEG-2-compliant slow-motion trick play	67
3.5	Conceptual plane-based MPEG-2-compliant video navigation	77
3.6	Networked hierarchical mosaic-screen navigation	80
3.6.1	Concept of hierarchical mosaic-screen navigation	81
3.6.2	System aspects of mosaic-screen navigation algorithm	82
3.6.3	Algorithm for MPEG-2-compliant mosaic-screen navigation	90
3.7	Experiments and validation results of both navigation concepts	94
3.7.1	Functional block diagram of Personal Video Recorder (PVR)	95
3.7.2	Performance measurement of implemented fast-search navigation	98
3.7.3	Performance measurement of implemented fast-search navigation	99
3.7.4	Performance estimation of slow-motion navigation	101
3.7.5	Performance estimation of navigation processing during recording	103
3.7.6	Picture quality validation of mosaic screens	104
3.8	Discussion on automated mosaic screen scrolling	109
3.9	Conclusions	110
4	Audio-enhanced dual-window video navigation	115
4.1	Introduction	115
4.2	Background and system aspects	117
4.3	Concept of audio-enhanced dual-window video navigation	122
4.3.1	Temporal subsampling of dual-stream video signal	123
4.3.2	Conceptual solutions for audio-enhanced dual-window video navigation	124
4.4	System integration and implementation	128
4.4.1	AV algorithm implementation of chosen navigation concept	128
4.4.2	Functional block diagram of the chosen concept	130
4.5	Computational reduced H.264/MPEG4-AVC intraframe decoding	133
4.5.1	Partial reconstruction of the prediction block	134
4.6	Experimental results	137
4.6.1	Picture quality of scalable MPEG-2 SD video decoding	138

4.6.2	Picture quality for H.264/MPEG4-AVC decoding concept 1 & 2	138
4.6.3	Computation reduction for H.264/MPEG4-AVC decoding	146
4.6.4	Test panel results on audio-enhanced dual-window navigation	147
4.7	Conclusion	149
5	Robustness improved DVB-H link layer	153
5.1	Introduction	153
5.2	DVB-H link layer essentials	154
5.3	Conceptually improved DVB-H link layer	160
5.4	Enhanced data recovery for an improved DVB-H link layer	162
5.4.1	Solution approach	162
5.4.2	Algorithm for IP recovery in defect MPE-FEC frame	167
5.5	Implementation and performance evaluation of the improved DVB-H link layer	175
5.5.1	Improved DVB-H link layer framework	175
5.5.2	Validation test set-up for DVB-H link layer	177
5.5.3	Algorithm performance validation	179
5.5.4	DVB-H link layer hardware test set-up	184
5.5.5	Performance results of the verified improved hardware link layer	186
5.6	Conclusions	189
6	Block-based detection systems for visual artifact location	193
6.1	Introduction	193
6.2	Background and related work	196
6.3	Conceptual artifact-location detection and filtering solution	201
6.4	Block-based artifact-location detection algorithms and low-pass filter control	205
6.4.1	Algorithm for spatial block-based artifact-location detection	205
6.4.2	Algorithm for frequency-domain artifact-location detection	210
6.4.3	Algorithm for control of the filter strength by entropy- based low-pass filtering	219
6.5	Experimental results of block-based artifact-detection	223
6.5.1	Evaluation of the detection approaches	225
6.6	Conclusions	244
7	Conclusions	247
7.1	Conclusion of the individual chapters	247
7.2	Discussion on research questions	250

7.3 Discussion and outlook	255
Appendices	257
A MPEG-2 Adaptation field	259
B MPEG-2 Timestamps	261
C MPEG-2 Section Syntax	263
D Characteristic Point Information	267
E Artifact-location detection on full-HD upscaled video	269
Publication List	271
Complete Bibliography	273
Acronyms	287
Acknowledgements	291
Curriculum Vitae	293

“Wer das Unmögliche nicht versucht, wird das Mögliche nie erreichen.”

Hermann Hesse, 1877 – 1962

Introduction

1.1 Preliminaries

Communication is a process where information is shared in space, time or a combination of them. Examples of space-based communication are e.g. telephone, radio and television, whereas time-shifted communication occurs when the information is recorded and played back at a later time instance, whereby the amount of delay depends on the application. Figure 1.1 depicts a basic system setup for video broadcasting using a satellite-based communication channel. At the left-hand side, cameras capture the scenery whereby the final video information is selected by the video mixer/editor. In order to achieve cost-effective broadcasting, the audiovisual signals provided by the mixer/editor are compressed, thereby potentially introducing visible coding artifacts in the video information signal, which negatively influences the final picture quality. The compressed audiovisual information signals are packetized and multiplexed into a packet-based signal, suitable to be transmitted across an error-prone channel. For static reception, the packets forming the multiplex are equipped with redundancy by the channel encoder, which adds block-code-based Forward Error Correction (FEC) data, extending the deployed packet length, enabling error detection and potential correction of erroneously received information. For mobile handheld-based communication, a second FEC layer may be required to secure the fidelity of the received information. The FEC equipped signal is finally modulated, using a modulation scheme matching the transmission channel characteristics. Besides space-based communication, Fig. 1.1 also depicts various time-based communication forms. A first form is located near the video mixer, enabling long-term and short-term storage of the captured scenery. An example of long-term storage is the generation of a Digital Versatile Disc (DVD) (pre-recorded video), whereas an example of short-term storage is the replay function, which is typically deployed during broadcasted live events. The compression deployed differs for both previous storage applications, as a result of different requirements, i.e. frame-accurate

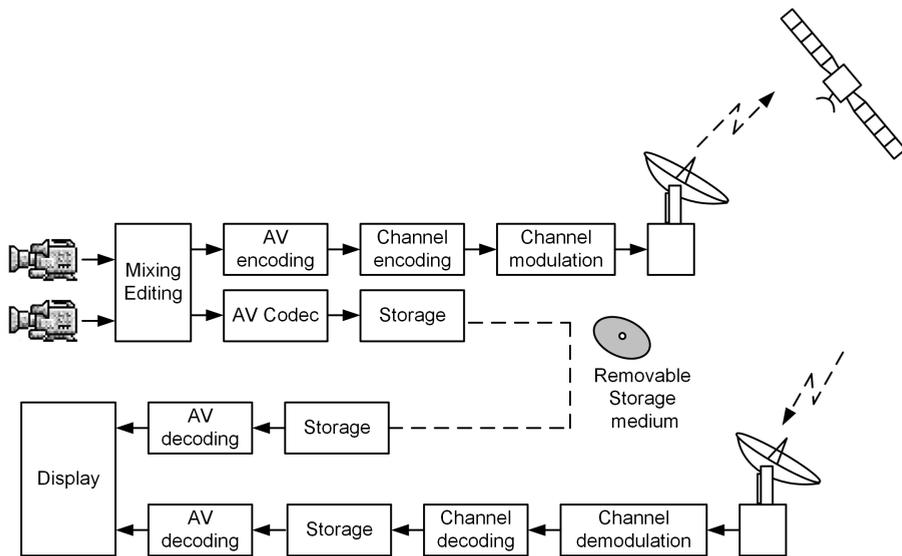


Figure 1.1 — *Video communication in time and space.*

video mixing versus high-quality high-volume storage. Note that pre-recorded video is an example of space- and time-based communication, as it is not only played back at a different time instance but, most probably, also at a different place. Another form of time-based communication is local storage at the broadcast receiver side, enabling consumers to view a broadcasted program at a later time instance.

All storage devices have means for video navigation, which is defined as video playback in non-consecutive or non-chronological order, compared to the original chronological capturing order. Video navigation can be divided into traditional fast-forward or fast-backward playback and advanced search methods, which are modern forms of video navigation. The former is found in analog and digital video recorders. The latter has become possible for random-access media such as disc and silicon-based memories. However, besides random access to the storage media, another important aspect is the applied video compression standard, which influences the video navigation.

Modern multimedia communication embraces the Internet Protocol (IP) for various reasons, such as robustness against latency or robustness regarding data duplication. Although the IP protocol is robust against data duplication, this aspect needs to be handled with care for battery-powered mobile communication systems deploying the IP protocol. This especially holds when designing a multi-chip receiver solution, whereby IP datagrams are forwarded to the

network layer via an external interface. In DVB-H, such a situation may arise when an additional FEC is available in the data link layer.

Cost-effective video broadcasting or storage involves video compression, which not only removes irrelevance but also relevant video information. As a result of this, visual coding artifacts are introduced, appearing in various forms such as blockiness, ringing, mosquito noise and contouring. Suppression of visible artifacts requires video post-processing typically involving a Low-Pass Filter (LPF) operation. In order to avoid image blurring, such an LPF is applied in a locally-adaptive manner. Temporal noise reduction not only attenuates Gaussian noise but, up to a certain extend, also non-static coding artifacts such as mosquito noise. In order to attenuate static visible mosquito noise and ringing, artifact-location detection followed by adaptive LPF is required. As coding artifacts differ in nature, so is their detection. The detection is simplified when considering only the contamination that occurs in flat or low-frequency regions as coding artifacts occurring in texture region are typically masked by that texture.

1.2 Background

In the past decade, advances in various technology fields enabled in Europe an industry-led consortium, known as the Digital Video Broadcast Project (DVB) [1], to specify a digital video broadcast system, which replaces the various analog-based broadcast standards. Note that in the US this was conducted by an industrial consortium group called the Grand Alliance [2]. Using open standards, the DVB Project specified the physical layer and data link layer, describing satellite (DVB-S), cable (DVB-C) and terrestrial (DVB-T) distribution systems for digital audiovisual information, data and associated return channels. After establishing these three communication standards, the DVB Project continued the development of new standards, replacing or enhancing existing standards. Digital Video Broadcasting Handheld (DVB-H) is such an enhancing standard forming a super-set of the DVB-T standard and targeting mobile handheld television reception using battery-powered devices.

With the introduction of DVB, the industry responded with the development of new products serving the needs of digital communication, thereby preserving existing features found in analog products, but also enabled new features, that where previously not possible in a cost-effective manner. This thesis presents improvements for MPEG-based consumer electronic systems in three non-overlapping areas, indicated by the gray-shaded blocks see Fig. 1.2. The three areas cover techniques and methods for video navigation in digital storage, robust and efficient link layer processing for DVB-H and reliable location detection of potentially mosquito noise and ringing contaminated regions.

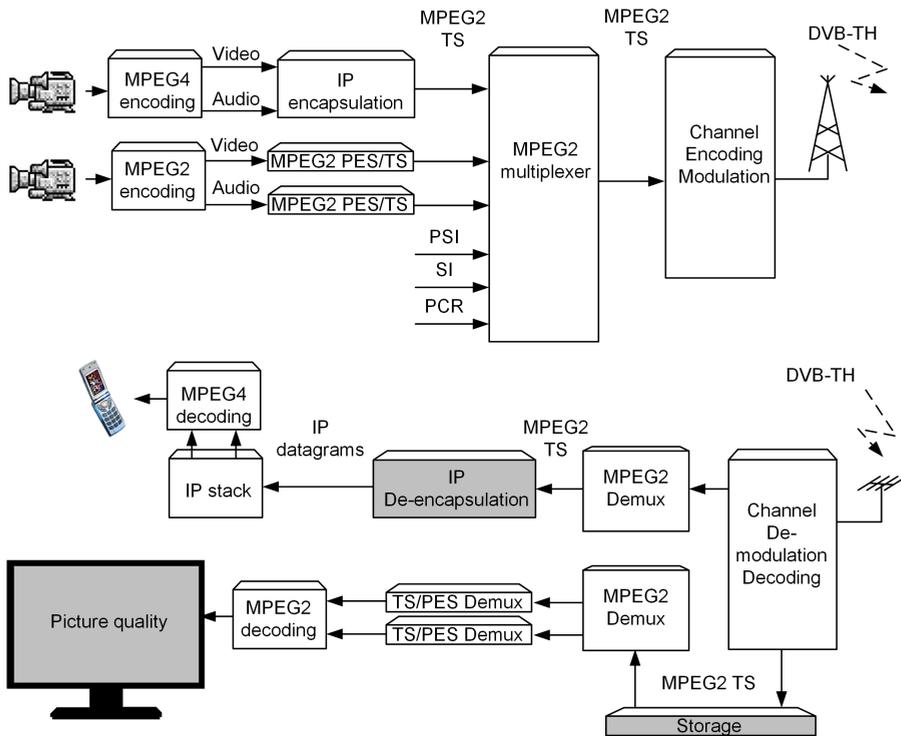


Figure 1.2 — Overview area of improvements.

A. Trick play for digital consumer storage systems

Digital video for consumer applications became available in the early 1990s, enabled by the advances in video compression techniques and associated standards and their efficient implementation in integrated circuits. Video compression deploying only transform coding resulted in the Digital Video DV [3] standard, whereas the convergence of transform coding and motion-compensated prediction into a single hybrid coding scheme, resulted in various standards in the early 1990s, such as MPEG-1 [4] and MPEG-2 [5]. These standards are capable of compressing video at different quality levels with modest up to high compression ratios and varying complexity. Each of the previous standards has been deployed in a digital storage system, based on different storage media. Hereby, a distinction is made between random-access and non-random-access storage media. Examples of the former are: optical disc, hard disk drive and solid-state memory, whereas an example of the latter is magnetic tape. Standards such as Video Compact Disc (VCD), Super Video Compact Disc (Super-

VCD or SVCD), Digital Versatile Disc (DVD or Blu-ray Disc (BD) make use of locator information to guide the system when performing fast-search playback. They exploit independently decodable pictures as entry points in the stream, where these pictures are also termed as intraframe-coded pictures. Tape-based storage devices require data processing during record, in order to support fast-search playback modes, whereby either the normal-play information is shuffled prior to storage on tape (DV), or by defining an extra, virtual channel, which is employed during fast-search playback (D-VHS), containing the fast-search information signal for a particular speed-up factor. The aforementioned storage systems either deploy a push- or a pull-based architecture. For the situation that a push-based architecture is deployed, which is the case e.g for D-VHS, the information stream retrieved from the storage medium, must be compliant with to the deployed compression standard in order to avoid buffer violations. The advantage of a pull-based decoding architecture is that it simplifies trick-play playback, resulting in lower system costs. An advantage of a push-based architecture is that the decoder can be separated from the storage system, enabling remote decoding of audiovisual information.

B. DVB-H mobile battery-powered handheld television

Robust handheld battery-powered television reception resulted in the development of DVB-H, a time-sliced broadcasting standard based on the Internet Protocol (IP). In order to provide sufficient robustness against channel impairments, the physical and data link layer are equipped with additional features, making DVB-H a super set of DVB-T. One of the additional robustness features, although optional, is an additional Reed-Solomon (RS)-based Forward Error Correction (FEC), which is located in the data link layer. This additional FEC protects the transmitted IP datagrams against various channel fluctuations, such as Carrier-to-Noise (C/N), Doppler and impulse interference. The data link layer FEC requires correctly and incorrectly received IP datagrams to be stored in memory, prior to FEC calculation. For the situation that one or more IP datagrams are incorrectly received, FEC calculation is required for trying to correct the erroneously received IP datagrams. For the situation that the FEC corrects all errors, all IP datagrams are forwarded to the network layer, whereby the IP datagrams are retrieved from memory using the IP datagram *total length* field, which is part of the IP header. However, when the FEC cannot correct the errors, all correctly received data may be lost, as the readout mechanism based on the IP datagram *total length* field may be corrupted. This data loss can be avoided, when correctly received IP datagrams are stored in FEC memory and also forwarded directly to the network layer. When after FEC calculation all erroneous IP datagrams are corrected, all IP datagrams are successfully forwarded to the network layer, resulting in data duplication of the correctly received IP datagrams. Although IP-based communication is robust

against data duplication, duplication should be avoided in a battery-powered receiver system for optimizing the power consumption. Without proper precautions, optimization of the power consumption may jeopardize the picture quality severely, due to the fact that audiovisual information is received in a time-sliced manner.

C. Picture quality for digital television

Bandwidth consumption in audiovisual broadcasting is dominated by the video signal. When deploying modern transform-based motion-compensated video compression techniques, cost-effective digital video broadcasting is achieved. Video compression not only removes irrelevant but also relevant information when quantizing transform coefficients, which effectively reduces the bit cost of pictures. This quantization not only deteriorates the picture quality due to lack of sharpness, but also introduces visible artifacts, which depend on the block size of the deployed video compression transform. Typical coding artifacts are blocking, contouring, ringing and mosquito noise, which negatively influence the perceived picture quality. As coding artifacts differ in nature, so is their detection. For mosquito noise and ringing, these coding artifacts may either be clearly visible or masked due to surrounding texture. For the situation that the mosquito noise and ringing artifacts are clearly visible, artifact attenuation involves locally-adaptive low-pass filtering, which can preserve the intended texture and avoid undesirable image blur. Due to the absence of a single metric revealing the presence and location of these artifacts, reduction of coding artifacts requires a two-stage approach, involving an artifact-location detection stage followed by a carefully controlled locally-adaptive low-pass filtering stage.

1.3 Research scope and problem description

This section delimits the scope of our research and details the research questions and design requirements at the system level for the three research fields.

1.3.1 Video Navigation for Digital Recording

The requirements for video navigation within MPEG-coded information are similar compared to conventional trick-play systems as found in digital storage media (disc, tape, etc). However, due to the fact that the video information is stored in compressed form, smooth video navigation requires more signal processing. For the consumer domain, it is essential to develop cost-effective navigation algorithms, which can be efficiently implemented in dedicated hardware or executed in software on either a Digital Signal Processor (DSP) or on the storage systems control processor. Furthermore, unlike tape-based storage

systems providing sequential access to the stored information, non-tape-based storage media allow random access. Modern storage systems allow random access to the compressed information, enabling high-speed search, but it is more complicated to find the useful information. In order to study cost-effective video navigation methods for digital MPEG-based audiovisual storage devices, the following research questions (RQ) are addressed in this thesis.

- **RQ1 How to efficiently perform trick-play playback on MPEG-compressed audiovisual information in various communication situations?**
- **RQ1a** How can normal-play MPEG-compressed audiovisual information be re-used for conventional trick-play playback?
- **RQ1b** How to perform trick play in a client-server-based networked system setup?
- **RQ1c** How to fulfill the bit-rate and frame-rate constraints when re-using normal-play MPEG-compressed video information?
- **RQ1d** What are the relations and limitations of high-speed search in relation to the MPEG-based playback navigation information?
- **RQ1e** What is the impact of the employed video format in relation to trick-play playback?
- **RQ1f** How can audio information contribute to the video navigation efficiency?
- **RQ1g** Is there a system architecture that allows conventional as well as more advanced video navigation methods?

1.3.2 Efficient and Robust DVB-H Link Layer

The DVB-H standard addresses IP-based DTV reception for handheld battery-powered receivers, where the DTV information is broadcasted in MPEG format. To increase reception robustness, the DVB-H link layer is equipped with an optional Reed-Solomon Forward Error Correction (RS-FEC) to correct erroneously received data. However, despite its error-correcting capabilities, the Reed-Solomon FEC is embedded in such a way that data duplication is required, leading to inefficiency and higher bandwidth requirements. In order to improve the DVB-H link layer on efficiency and robustness, additional processing is required. This study leads to the following research questions (RQs) and system requirements (SRs) that need to be addressed.

- **RQ2 How to improve the robustness of a standard DVB-H link layer while avoiding excessive load on system resources?**

- **RQ2a** How can the error recovery of the embedded RS decoder be optimized leading to improved robustness?
- **RQ2b** How to communicate correctly received and FEC-corrected IP datagrams in a smooth communication way?

1.3.3 Block-based Visual Artifact-Location Detection

Cost-effective MPEG-based video broadcasting is inherently operated at a low bandwidth to save communication costs. Consequently, the MPEG compression system, employing a block-based Discrete Cosine Transform (DCT) and subsequent strong quantization of the DCT signal components, introduces coding noise in the video signal. When classifying the coding noise in more detail, mosquito noise and ringing are the most annoying, especially when they occur in flat and/or low-frequency regions. For the situation that these artifacts occur in a dynamic fashion, temporal-noise filtering typically attenuates this distortion. However, for the situation that these artifacts are static, a different solution is required. To facilitate detailed noise removal on the locations where the noise is annoying, artifact-location detection is required, which reveals the locations that potentially contain visible noise patterns, so that they can later be removed with selective filtering. This issue involves the following research questions (RQs).

- **RQ3** How to efficiently detect visible coding noise locations in MPEG-coded video with sufficient performance?
- **RQ3a** With what methods can visible MPEG noise patterns reliably be found in the image and what are the corresponding metrics?
- **RQ3b** How can the reliability of the detection methods be improved?
- **RQ3c** How can this method be embedded in a DTV platform?

1.4 Contributions of the research

This thesis presents improvements for different types of MPEG-based consumer communication systems. The conducted research covers three MPEG-related areas: navigation techniques for client-based communication and storage systems, an efficient and robust DVB-H link layer for mobile television-reception and visual artifact-location detection for image enhancement in digital television communication.

A. Navigation for client-based communication and storage systems

This thesis presents three video navigation methods for MPEG-based video communication, each addressing different navigation criteria. The first video navigation method aims at fast-search and slow-search trick play on the basis of re-used normal-play video information, generating an MPEG-compliant information signal, which is suitable to be used in a client-server-based network architecture. The second video navigation method introduces a new hierarchical mosaic-screen-based video browsing method for networked communication, presenting an instantaneous overview on the basis of a set of re-usable images. This is derived from a particular normal-play time interval, which depends on the employed hierarchical navigation layer. The mosaic screens form in combination with predictive-coded images an MPEG-2-compliant signal, suitable to be used in a client-server-based network architecture. The third video navigation method aims at providing a multi-signal navigation scheme, containing both audio and video information. The method deploys normal-play audiovisual fragments in combination with fast-search video information, which is simultaneously presented within a dual-window video screen. In this way, the navigation signal simultaneously uses the human visual and auditory cues, thereby making the scene more informative for navigation purposes, while enabling the user to perform other tasks in parallel.

B. Efficient and Robust DVB-H Link Layer

We have presented a method to improve the efficiency and robustness of an MPEG-based DVB-H receiver, while providing a best-effort signal degradation. In a standard approach of the DVB-H link layer, correctly received or RS-FEC-corrected datagrams cannot be retrieved from a defect data frame after reception. This leads to unnecessary data duplication and communication bandwidth. We propose a solution that is based on locally obtained reliability and location information, facilitating an improved FEC performance and the ability to locate correctly received and locally-corrected IP datagrams. Knowledge on the location of correct IP datagrams allows them to be communicated only once to the network layer, thereby considerably reducing the involved data traffic and handling. The robustness is improved by deriving the involved reliability information on the basis of small data packets, thereby improving the balance between correct, and incorrectly received IP datagrams. By implementing our scheme, we not only improve the robustness and efficiency of MPEG-based DVB-H reception, but also enhance the power consumption considerably. This improved DVB-H link layer has been embedded in a commercial chip.

C. Detection of Visual Coding Artifact Locations

MPEG-compressed video communication is typically performed with limited bandwidth to save costs, at the expense of video coding noise in the reconstructed images. We propose two artifact-location detection methods, either operating in the spatial domain or frequency domain, for the location detection of mosquito noise and ringing noise patterns. Both solutions involve a block-based detection kernel and calculate an activity metric on a per block basis, which is divided for later filter control, using a simple classification. By employing context reasoning within the detection kernel, involving the block-based signal features, a distinction is made between potentially contaminated and intended texture. For the situation that the a potentially contaminated region is detected, the detection is two-dimensionally extended, involving the block-based signal features. We have found that the spatial-domain solution always provides a locally enhanced PSNR for a broad range of image data and compression ratios, while the frequency-domain solution shows a fluctuating performance which only occasionally outperforms the results obtained in the spatial domain. The solution operating in the spatial domain has been successfully embedded in three commercially available DTV chips for the reduction of static noise patterns.

1.5 Outline and scientific background

This section presents an outline of the chapters and briefly elaborates the key contributions of the individual chapters. The scientific contribution of each chapter is listed per chapter on the basis of the realized publications. The thesis outline is depicted in Fig. 1.3. Chap. 2 provides an introduction to the deployed compression schemes and protocols used by DVB. The actual research contributions are split over four chapters, Chapters 3– 6. Chapters 3 and 4 discuss three forms of video navigation. Chapter 5 presents a robust and efficient DVB-H link layer. Chapter 6 addresses the work on visual coding artifact-location detection and reduction. In Chapter 7, conclusions on the conducted work are presented.

In Chap. 3 we discuss two solutions for MPEG-2-compliant trick play, which is required when connecting a digital TV to a remote video server. In Chap. 4 another novel video navigation method is presented based on a dual-window video concept, combining normal-play video fragments with associated sound and a fast-search video navigation signal. Chapter 5 introduces a robust and efficient DVB-H link layer, deploying a concept that derives reliability information during reception and forward error correction, enabling to forward correctly received IP-based information is only forwarded once to the IP-stack. Chapter 6 presents two block-based detection systems, for locating visual mos-

quito noise and ringing artifacts, enabling visibility reduction on the basis of locally-adaptive low-pass filtering.

Chapter 2 presents the MPEG technology, which forms the basis of the conducted work and is common to all chapters. The chapter also contains a brief description of the encapsulation of MPEG-compressed data in the MPEG-2 Transport Stream, utilized in the DVB-H broadcast standard for mobile video communication. Furthermore, a brief introduction is given on digital storage and associated techniques for video navigation.

Chapter 3 addresses two video navigation solutions suitable for a client-server-based communication setup. The first method is based on conventional fast-search and slow-motion video playback for digital recording systems. The second video navigation method involves mosaic screens, constructed from images derived from the normal-play video sequence, either obtained via uniform subsampling or based on specific filter criteria, enabling video browsing

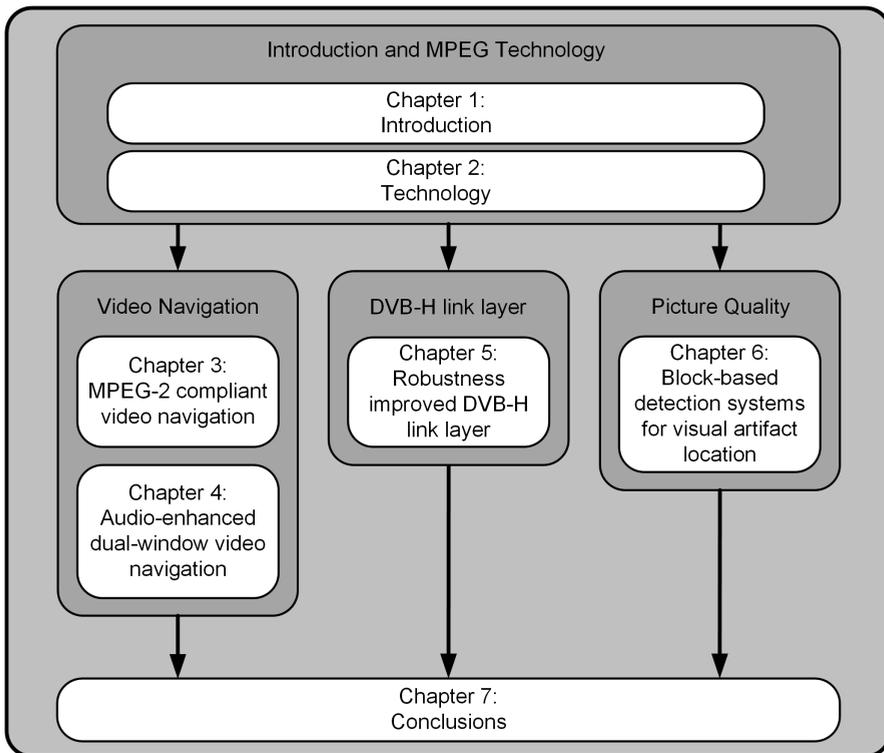


Figure 1.3 — *Structure of the thesis.*

in a hierarchical manner. To facilitate interoperability between the client and the server in a communication setup, a special algorithm is presented solving the trade-off between bit rate and frame rate within the boundaries of the MPEG standard to generate a compliant trick-play information signal. The contents of this chapter are presented in 4 patent applications: US6400888B1 [6], US2002/0167607A1 [7], US2006/0050790 [8] and US2003/0231863 [9] and 2 contributions to the IEEE Int. Conf. on Cons. Electr. (ICCE) 2002 and 2003 [10], [11]. Two journal papers were published in the IEEE Trans. on CE [12], [13]. Furthermore, this work was also released as part of a book chapter [14].

Chapter 4 extends the work on video navigation by adding audio as additional navigation signal. This leads to audio-enhanced video navigation, deploying dual-window-based trick play, utilizing normal-play fragments shown in a main window, in combination with a fast-search information signal rendered in a Picture-in-Picture (PiP) window. The results are covered in a patent application: US2007/0035666A1 [15] and a contribution to the IEEE ICCE 2008 [16]. A journal paper was published in the IEEE Trans. on CE [17].

Chapter 5 introduces an efficient and robust DVB-H link layer, using both correctly received and error-corrected data and the associated reliability information before and after FEC, to enhance the data robustness and communication efficiency. We present an algorithm to communicate the IP datagrams only once to the network layer. The link layer robustness is improved by carefully assigning reliability information to the received data, which is later exploited to establish Quality-of-Service (QoS). The work of this chapter is based on 4 patent applications: US2008/0209477 [18], US2008/0282310 [19], US2009/0006926 [20] and US20080263428 [21]. Furthermore, it was published in the IEEE ICCE 2006 [22], [23] and 2007 [24]. It was also published in 2 journal papers to the IEEE Trans. on CE [25], [26]. It was additionally published as a book chapter [27].

Chapter 6 contributes to the detection methods for visual MPEG coding-noise locations and elaborates on two artifact-detection kernels using spatial-domain and frequency-domain representations. For both methods and on the basis of an *activity* metric classification, a local video model is derived suitable for context reasoning, enabling the identification of potential artifact contamination. The derived detection methods are both evaluated using a locally-adaptive low-pass filter, with which actual local video quality could be improved. This work is published in IEEE ICCE 2013 [28] and in the IEEE Trans. on CE [29].

Chapter 7 discusses the obtained results and combines the methods for video navigation. The results for the improved DVB-H link layer are put in a perspective for mobile communication. It is argued that the artifact-location detection is also applicable for future video-compressed communication systems.

Technology overview

2.1 Introduction

In 1988, the Motion Picture Expert Group (MPEG) working group, with official name ISO/IEC JTC1/SC29/WG11, was founded with the objective to define compression standards for audiovisual information. The first standard was MPEG-1 [4], [30], [31], which became available in 1991 and formed the basis for other standards such as Video CompactDisc (VCD) and Digital Audio Broadcasting (DAB), or related products such as MP3-based music players. Limitations of the MPEG-1 Video standard, like support for interlaced signals and the lack of a robust transport stream format made the deployment of this standard unsuitable for Digital Video Broadcasting (DVB). The shortcomings of the MPEG-1 standard were solved by its successor MPEG-2, which became available in 1994, paving the way for DVB and also for the successors of VCD like Super Video Compact Disc (SuperVCD or SVCD), Digital Versatile Disc (DVD) and it initially also formed the basis for Blu-ray Disc (BD).

At the end of the 1980s, industry was working on a successor for the various analog-based television broadcast standards. In the USA, this resulted in forming the Grand Alliance (GA), a consortium of companies developing the American HDTV standard, while Europe established an industrial consortium known as the Digital Video Broadcasting (DVB) Project. Due to differences between the American and European broadcasting business models and associated industry influences, the digital broadcasting standards targeting the USA and Europe are different on certain aspects. Common in both broadcast systems is the usage of the MPEG-2 Video compression standard and the way of multiplexing compressed audiovisual information, described by the MPEG-2 Systems standard.

The DVB Project used the MPEG-2 standard to develop a range of broadcasting standards, utilizing international open standards, addressing distribution via terrestrial, cable and satellite channels, also known as DVB-T, DVB-C and DVB-S, respectively. Since the MPEG-2 Video standard was well equipped with television broadcasting *Profiles* and *Levels* and interlacing was well

covered, this standard was completely adopted and integrated into DVB. Although the MPEG-2 Systems standard plays a key role in DVB, the standard does not cover all aspects for constructing a robust digital broadcasting system. For this reason, the DVB Project specified the physical layer, data link layer and associated return channels. The Program Specific Information (PSI) specified by MPEG-2 is not sufficient to enable e.g. an automatic adjustment of end-user equipment. Therefore, the DVB-Project introduced Service Information (SI) to solve the MPEG-2 Systems shortcomings [32]. After establishing the DVB-S/T/C communication standards, the DVB Project continued the development of new standards, replacing or enhancing existing standards. Digital Video Broadcasting Handheld (DVB-H) is such a new standard, which was enabled by a new video coding standard known as H.264/MPEG-4 AVC.

With the appearance of DVB, customers expect new products with higher quality and features that are more sophisticated compared to features known from analog systems. Digital storage products such as Personal Video Recording (PVR), allow manufacturers to develop specific features in the area of trick play and video navigation, which distinguish their products from traditional systems. Such a specific feature is video browsing, which is also known as trick play or intra-program navigation. However, such a feature should be feasible with low costs and minimal impact on the PVR bill of material. This requires cost-effective signal processing, typically mostly executed on the embedded platform DSP or control processor.

The low output bit rate of H.264/MPEG4-AVC has enabled mobile handheld-based television. Although this standard separates the video coding layer from the network layer enabling a more robust transmission, this robustness is not sufficient for handheld television. Specific standards addressing handheld broadcasting, such as e.g. DVB-H [33], [34] therefore provide additional robustness by employing an additional Forward Error Correction (FEC) stage to protect the IP-datagram-based broadcast data. To facilitate the readout of correct or corrected IP datagrams, a special mechanism is deployed, which is not part of the DVB-H standard, thereby contributing to a robust and efficient system implementation.

Cost-effective broadcasting involves lossy video compression, whereby not only irrelevant but also relevant information is removed from the video information, resulting in visual coding artifacts. MPEG-2 coded video broadcast signals typically suffer from various impairments, while modern video coding standards such as H.264/MPEG4-AVC have means to reduce particular coding artifacts. In order to reduce such artifacts, coding-artifact reduction as a video post-processing step remains required in modern digital television systems. Coding artifacts differ in nature, which is confirmed by the various proposals to detect and reduce these artifacts [35]. Two closely related artifacts are mosquito noise and ringing. These artifacts occur due to the removal of frequency information by the quantization process deployed by the video encoder. The

visibility of these artifacts depends on the nature of the spatial video. Both artifacts are typically noticeable in flat and/or low-frequency regions, preceding or succeeding the texture or edge region. Detection of these transition regions, either in the time domain or frequency domain, provides location information, from which a succeeding filtering stage can benefit, as the discrimination between intended texture and artifact contamination is improved. Such a filtering solution is an example of locally-adaptive processing to reduce coding artifacts, while minimizing image blur.

2.2 MPEG-2 Standard

MPEG-2 is an international standard and is officially referenced as ISO/IEC 13818. The video and systems part of this standard was developed in collaboration with ITU-T and therefore also listed as ITU-T Recommendation H.262 for the video part and ITU-T Rec. H.222 for the systems part. The ISO/IEC 13818 standard consists of 11 parts¹, each describing a specific part of the standard, see Table 2.1. The parts from Table 2.1 which are relevant for this thesis are 1, 2, 3, 5, 6 and 9.

2.2.1 MPEG-2 Part 2: Video

The MPEG-2 Video specification deploys a semantic in combination with an associated syntax, which is generic and serves a wide range of applications, such as digital storage and television broadcasting. In order to enable a cost-effective practical implementation, the syntax is constrained using *Profiles* and *Levels*. *Profiles* define a subset of the entire syntax, while *Levels* constrain certain values of the allowed syntax forming the coded bit stream. MPEG-2 video compression is a block-based hybrid video coding scheme as depicted in Fig. 2.1. From Fig. 2.1 it becomes clear that MPEG-2 encoding is more complex than MPEG-2 decoding, which is typically the case for many classes of video coding. Block-based video compression schemes operate on a group of spatially adjacent pixels. A distinction is made between intraframe and interframe compression. In *intraframe* compression, only spatial information is used for compression, whereas for *interframe* compression also information from the past and or the future pictures is used for compression, see Fig. 2.2(b). The advantage of intraframe compression is that the picture can be decoded independently, thus without the need of information from previous or future pictures, thereby providing random access into a video sequence. However, a drawback is the modest compression factor, typically requiring a transmission time

¹Part 8 has been dropped due to lack of industry interest in 10-bit video.

Table 2.1 — *International standard ISO/IEC 13818 and sub-standards.*

Part	Number	ITU-T Rec.	Title
Part 1	ISO/IEC 13818-1	H.222.0	Systems
Part 2	ISO/IEC 13818-2	H.222.0	Video
Part 3	ISO/IEC 13818-3		Audio
Part 4	ISO/IEC 13818-4		Conformance testing
Part 5	ISO/IEC 13818-5		Software simulation
Part 6	ISO/IEC 13818-6		Extensions for DSM-CC
Part 7	ISO/IEC 13818-7		Advanced Audio Coding (AAC)
Part 8	ISO/IEC 13818-8		10-Bit Video
Part 9	ISO/IEC 13818-9		Extension for real-time interface for systems decoders
Part 10	ISO/IEC 13818-10		Conformance extensions for Digital Storage systems decoders Control (DSM-CC)
Part 11	ISO/IEC 13818-11		IPMP on MPEG-2 Systems

of more than a picture display period, which is the reciprocal of the video frame rate. The main purpose of intraframe-compressed pictures is to facilitate random access to a video broadcast or stored video sequence. To achieve a higher efficiency in video compression, interframe compression needs to be part of the video coding scheme, as it involves the transmission of only the changes compared to a reference picture. In the extreme situation that two successive pictures are equal, the interframe-compression method ultimately requires only the transmission of information indicating that a picture is a duplicate of the previous or future reference picture. This duplication rarely occurs in practice, but it can occur in specific situations such as in trick-play playback.

MPEG-2 video compression is achieved by applying a Discrete Cosine Transform (DCT) to a group of 8×8 pixels, of which the resulting components are quantized and runlength coded. For the situation that the video originates from an interlaced video source, the video lines from top- or bottom-field can be separated, either at the macroblock level or at picture level, prior to transformation, creating a field-coded macroblock or a field-coded picture, thereby increasing the coding efficiency in case of significant motion. A macroblock consists of four luminance (Y) DCT blocks and two chrominance blocks (Cb,Cr)

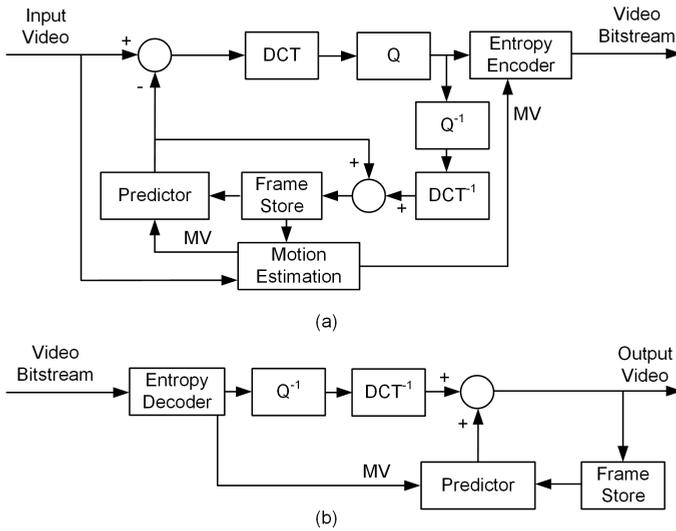


Figure 2.1 — MPEG-2 video codec. (a) Encoder. (b) Decoder.

in case of the usual 4:2:0 color subsampling format, see Fig. 2.2(a). To facilitate specific features in transmission and storage, a macroblock is equipped with horizontal locator information, also known as *macroblock_address_increment* which reveals, when added to the previous macroblock locator information, the absolute horizontal location of the macroblock. Typically, multiple successive macroblocks are combined into a slice, forming a horizontal row of macroblocks in the image. Such a slice contains a *slice_start_code*, indicating the vertical locator information, see Fig. 2.2(a). For interframe compression, a macroblock can be predicted from previous and/or future reference pictures. This type of prediction can result in a P-picture, when predicted from a single past reference picture, or a B-picture when predicted from two reference pictures (past and future), see Fig. 2.2(b), which can be frame- or field-based, see Fig. 2.3. If the temporal difference is large within the motion-estimation search area, a macroblock is intraframe-coded, even if the picture type is P-type or B-type coded². Figure 2.2(b) indicates a typical video coding situation for a 25-Hz broadcast situation, resulting in an intraframe distance of $N = 12$ and a uni-directional distance $M = 3$. This intraframe distance and the occurrence of an I-picture marks the border and start of a Group Of Pictures (GOP), allowing random access. Figure 2.2(c) indicates picture transmission reordering, required to fa-

²In the sequel of this thesis, we sometimes denote I, P, B-type pictures in abbreviated form as I-, P-, B-picture

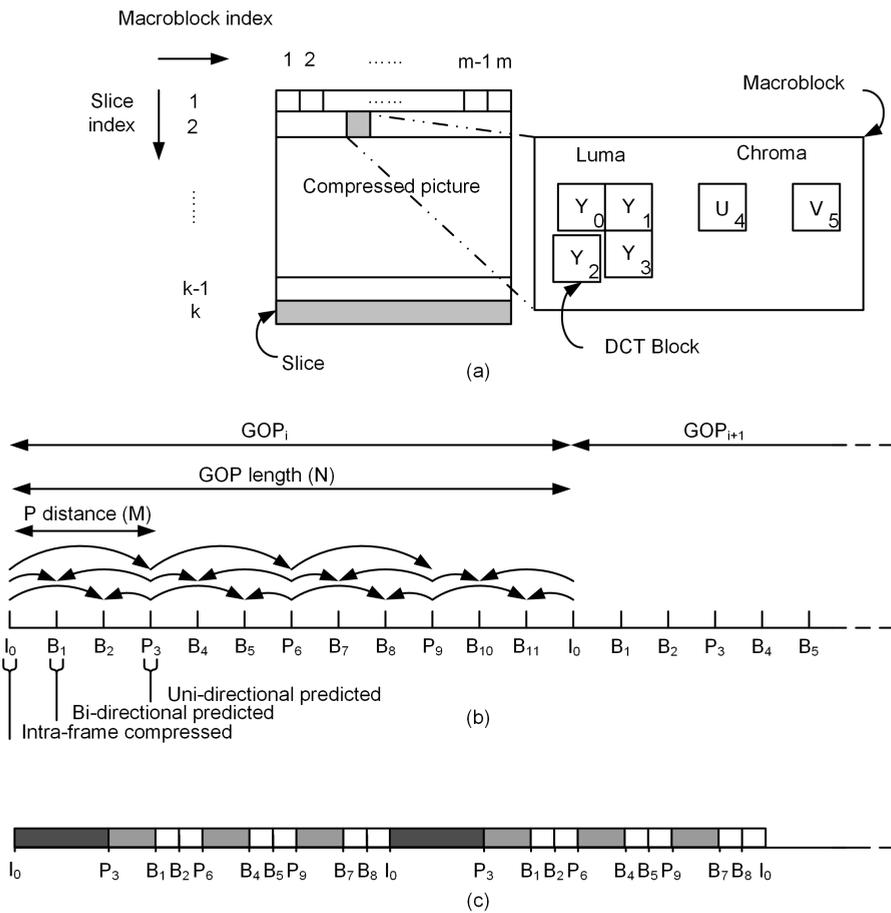


Figure 2.2 — MPEG-2 block-based hybrid video coding. (a) Block-based compressed picture and its main compression syntax elements for video with 4:2:0 color subsampling format. (b) Interframe prediction for a GOP structure with $M = 3$ and $N = 12$. (c) Reordering of MPEG-2 coded pictures.

cilitate decoding of bi-predicted pictures in a streaming manner. The example in Fig. 2.2(c) indicates an open GOP structure, whereby the decoding of bi-directionally coded pictures B_{10} and B_{11} of GOP_i depends on the availability of the intraframe-coded picture of GOP_{i+1} . In MPEG notation, the GOP length is indicated by N , while the distance between two successive P-pictures is indicated by M . For the situation that a GOP has a length of $N = 12$, this yields

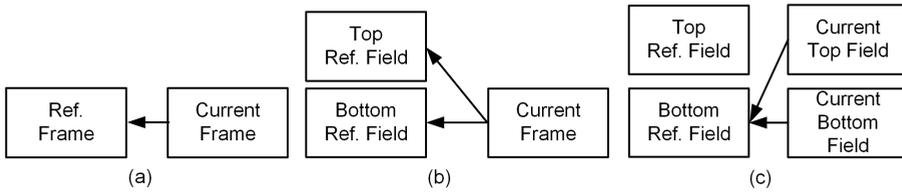


Figure 2.3 — *Examples of frame- and field-based prediction. (a) Frame-based prediction. (b) Top/bottom-field prediction. (c) Bottom-field prediction.*

a worst-case decoding latency of 480-milliseconds, which equals 12 frame periods for a 25-Hz television system.

2.2.2 MPEG-2 Part 1: Systems

Besides specification of compression standards for audiovisual information, MPEG has also specified a semantic and syntax that constructs a packetization scheme, enabling multiplexing of MPEG-compressed audiovisual information and data according to the ISO/OSI Layered model, see Fig. 2.4. The possible packetization/multiplexing methods are described by the standard ISO/IEC-13818 Part 1: Systems. The standard addresses two methods: packetization for either quasi error-free and error-prone channels, resulting in a Program Stream (PS) and a Transport Stream (TS), respectively. Practical examples of such channels are Digital Versatile Disc (DVD) for quasi error-free and Digital Video Broadcasting (DVB) for an error-prone channel. Both TS and PS provide a coding syntax, enabling synchronization, decoding and presentation of audiovisual information. These technical terms involve a number of specific measures in the standard and formats to facilitate the system aspects. For example, the coding syntax of TS and PS use packet headers with specific start codes to enable random access. Furthermore, synchronization is implemented by inserting timestamps into the packet streams, enabling synchronization of decoders and presentation process of the individual audiovisual streams. Figure 2.5 indicates the basic MPEG-2 encoding and packetization steps constructing an MPEG-2 TS. At the top of Fig. 2.5(a), uncompressed audiovisual information is compressed by their corresponding encoders, which generate an Elementary Stream (ES), as depicted in Fig. 2.5(b). The ES forms the input for the packetizer, which separates the ES in access units, which are decodable entities. For coded video, an access unit is e.g. a compressed picture, whereas for coded audio an access unit is a group of compressed consecutive audio samples, indicated as a frame. The packetizer generates a Packetized Elementary

Stream (PES), by adding a Packet Header (PH) to the compressed access unit, indicating e.g. when the access unit has to be decoded (DTS) and should be rendered (PTS), see Fig. 2.5(c) and Appendix B. Finally, the PES is partitioned into a stream of 188-Byte fixed-length packets, known as Transport Stream (TS) packets. During multiplexing, additional information is added to the Transport stream Header (TH) such as e.g. a Program Clock Reference (PCR), which is required during decoding for reconstruction of the decoder time-base, which enables proper decoding of the received audiovisual access units. Figure 2.6 depicts a basic MPEG-2 decoder operating on a TS, which is the output signal of a DVB channel decoder.

A. MPEG-2 Packetized Elementary Stream

In a Packetized Elementary Stream (PES), the ES access units are equipped with a header, which enables the transmission of various information fields, related to the access unit and its decoding process. The presence of these fields is indicated by binary flags in the PES header. For this thesis, two information flags are important: the *PTS_DTS_flags* and the *DSM_trick_mode_flag*, for indicating the presence of information regarding the timing for decoding and presentation, and the repetition behavior of pictures during trick play. More information on other flags can be found in [36]. More specifically, the *PTS_DTS_flags* indicates the presence of a *Decoding Time Stamp* (DTS) and the presentation time *Presentation Time Stamp* (PTS). For exact time stamp calculation, see Appendix B.

ISO/OSI layer	MPEG-2 layer		Layer	Definition
5: Session	Compression	Audiovisual encoding/decoding	7	Application layer
4: Transport	System	PES packet layer - Synchronizing individual information streams	6	Presentation Layer
		Transport packet layer - Multiplexing/demultiplexing - Buffer management - Timing	5	Session Layer
			4	Transport Layer
			3	Network Layer
			2	Data link Layer
			1	Physical Layer

(a)

(b)

Figure 2.4 — MPEG-2 Systems according to the ISO/OSI layering model. (a) MPEG-2 Layering model. (b) ISO/OSI Layer model.

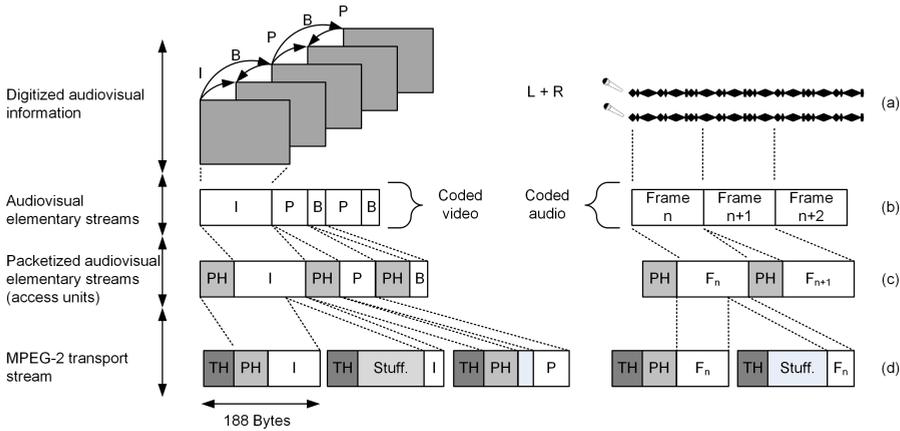


Figure 2.5 — MPEG-2 processing steps after compression. (a) Time-domain audiovisual information. (b) Compressed audiovisual access units. (c) Packetized audiovisual access units. (d) Segmenting audiovisual access units into MPEG-2 TS packets.

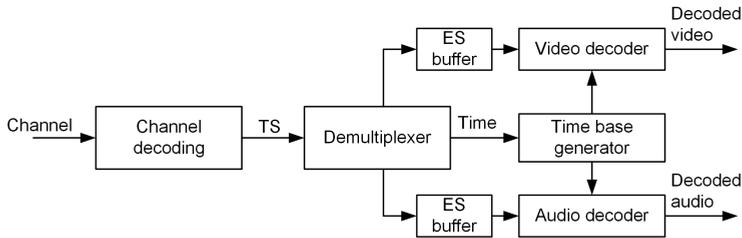


Figure 2.6 — Basic overview of MPEG-2 decoding.

The *DSM_trick_mode_flag* indicates the presence of information regarding trick-play playback modes, such as fast-search and slow-search at various playback speeds, or freeze-frame display. However, although the usage of trick modes associated with the *DSM_trick_mode_flag* is recommended for decoding systems equipped with a digital interface, this is not strictly demanded [37]. As a result, the support for these trick-mode facilities is not reliable so that other methods are required, circumventing the need for the *DSM_trick_mode_flag* and its associated playback, while obtaining similar or equal trick-play playback.

Syntax	Number of bits
transport_packet() {	
sync_byte	8
transport_error_indicator	1
payload_unit_start_indicator	1
transport_priority	1
PID	13
transport_scrambling_control	2
adaptation_field_control	2
continuity_counter	4
if (adaptation_field_control == '10') (adaptation_field_control == '11') { adaptation_field() }	
if (adaptation_field_control == '01') (adaptation_field_control == '11') { for (j=0; j<N1; j++) { data_byte } }	8
}	

Figure 2.7 — Syntax of MPEG-2 transport packet header.

B. MPEG-2 Transport Stream

The MPEG-2 Transport Stream (TS) is packet-based with a fixed length of 188 Bytes. The maximum payload capacity is 184 Bytes, resulting in a 2 % overhead due to the 4-Byte TS header. This 4-Byte TS header syntax is depicted in Fig. 2.7. A TS packet starts with an 8-bit *sync_byte* field, which has a fixed value of “0x47”, followed by three 1-bit flags. The *transport_error_indicator* flag, when set to “0x0”, indicates whether this TS packet has been received correctly, while when set to “0x1”, the TS packet contains errors. It is the responsibility of the channel decoder to set this flag based on the outcome of a Reed-Solomon Forward Error Correction (FEC) calculation [38]. For this, 16 Bytes are added to each TS packet by the channel encoder, extending the TS packet length to 204 Bytes. For the situation that the TS packet carries audiovisual data, the *payload_unit_start_indicator* flag, when set to “0x1”, indicates that the payload starts with the first Byte of a PES packet, while a “0x0” indicates that no PES data will be present in this TS packet. For the situation that the TS packet contains section data, the *payload_unit_start_indicator* flag, when set to “0x1”, indicates that the first Byte of the payload holds the pointer field, a syntax field that points to the location inside the current TS packet where the first Byte, i.e. the *Table_id*, of the section starts. If the *payload_unit_start_indicator* flag is set “0x0”, the TS packet does not contain the start of a new section. The flag *transport_*-

Table 2.2 — *Predefined MPEG-2 PID values.*

PID value	Description
0x0000	Program Map Table
0x0001	Conditional Access Table
0x0002, ..., 0x000F	Reserved
0x1FFF	Stuffing

priority reveals, when set to “0x1”, that this TS packet has a higher priority compared to other TS packets with the same PID value. The Packet Identifier (PID) can be regarded as the name of the transported information. It has a 13-bit value enabling 8,192 unique information streams to co-exist simultaneously in a single multiplex. Although the PID is based on a 13-bit value, not all values can be generally used, as some PID values are reserved for particular information streams, see Table 2.2. The 2-bit *transport_scrambling_control* flag, when set to “0x00”, indicates that the TS packet is not scrambled, whereas the values “0x0”, “0x10” and “0x11” indicate that the TS packet is scrambled using a method which is defined by the user. The 2-bit *adaptation_field_control* flag indicates whether the TS header is followed by an adaptation field, see Fig. 2.7. The TS packet header finishes with the *continuity_counter*, which is incremented with unity, for each TS packet with the same PID value and wraps around modulo 16. In principle, for a single service constructed of TS packets with the same PID value, the *continuity_counter* enables packet loss detection of 14 consecutive TS packets. Basically, this means that the burst length of lost TS packets that can be detected is $14k$, with k being the number of multiplexed services, provided that they are all time-interleaved corresponding to a round-robin multiplexing scheme i.e. sequential periodic counting.

An MPEG-2 TS header can be succeeded by an adaptation field, indicated by the *adaptation_field_control*. This adaptation field enables the transmission of various information signals such as the Program Clock Reference (PCR), see Appendix A. Furthermore, this adaptation field is used to insert stuffing data, which is required when audiovisual access units have insufficient data to fill the payload part of a TS packet, see Fig. 2.5(d) [36]. For the situation that section data is contained by a TS packet, this adaptation field may occur, however, insertion of stuffing Bytes is facilitated by stuffing the payload part of a TS packet with “0xFF” valued bytes, starting directly after the last section Byte.

2.2.3 MPEG-2 Part 3: Audio

Besides the development of video compression standards, MPEG has also developed audio compression standards. The first result was the ISO/IEC 11172-3 standard [31], which offered three layers of audio compression with increasing complexity, known as Layer I, II and III, capable of compression mono and stereo audio signals. The ISO/IEC 13818-3 standard is the MPEG-2 audio compression standard and forms an extension of the MPEG-1 audio compression standard. A major difference, compared to MPEG-1 audio compression, is the support of 5.1 multichannel audio, also known as 3/2 multichannel audio and an optional low-frequency enhancement channel (LFE). The LFE channel is capable of handling signals in the range from 15 Hz to 120 Hz. Furthermore, to increase the audio quality at low bit rate, three additional sampling frequencies F_s : 16 kHz, 22.05 kHz and 24 kHz have been introduced, while additional bit rates are possible. Finally, MPEG-2 Audio is encoded in such a way that although multichannel audio is encoded, this is always performed in such a way that a stereo information signal is available, enabling backward compatibility with existing MPEG-1 decoders.

Figure 2.8(a) depicts a basic block diagram of an MPEG-2 audio encoding scheme. At the left-hand side, Pulse Code Modulation (PCM) audio samples enter the *mapping* stage, which creates a filtered and subsampled representation of the input audio stream. The audio samples can either be mono, stereo or multi-channel. The mapped samples are either called subband samples, or transformed subband samples. The *psychoacoustic* processing block creates a set of data, which controls the quantization and coding process. The applied psychoacoustic model operates over a set of consecutive audio samples, whereby the amount of involved samples depends on the deployed coding Layer. Table 2.3 indicates the time duration of a compressed audio access unit, also known as *frame*. The *frame packing* stage assembles the final bit stream, multiplexing the compressed audio information, optional ancillary data and optional CRC-based error detection.

Figure 2.8(b) depicts a basic block diagram of an MPEG-2 audio decoding scheme. At the left-hand side, the bit stream enters the decoder. The *frame unpacking* block performs error detection, provided this feature is facilitated by the encoder and separates the bit stream. The *reconstruction* stage converts the quantized subband samples (mapped samples), which are, by means of inverse mapping transformation, converted into PCM samples. A *frame* is an independent decodable entity (access unit) and consists of 384 audio samples for Layer I and 1152 audio samples for Layer II and III. A *frame* starts with a so-called syncword, and finishes before the next syncword. It consists of an integer number of *slots*, whereby a *slots* consists of 4 Bytes in Layer I and a 1 Byte in Layer II. Layer III differs from Layer I and II in the sense that although the distance between consecutive syncwords resembles an integer amount of Bytes, the audio

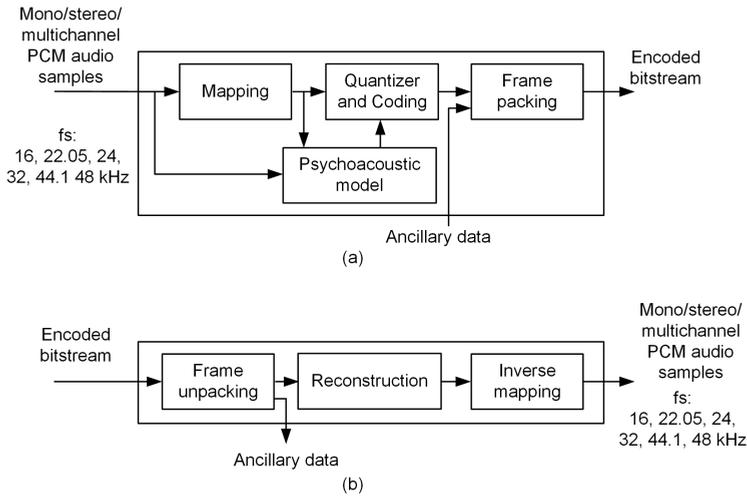


Figure 2.8 — *MPEG-2 audio coding. (a) MPEG-2 audio encoder. (b) MPEG-2 audio decoder.*

information that belongs to a *frame* is generally not contained between two successive syncwords. There are sample frequencies such as 44.1 kHz in MPEG-1 and 22.05 kHz in MPEG-2 audio compression, which require padding. Padding is a method to adjust the average length of an audio frame in time to the duration of the corresponding PCM samples, by conditionally adding a *slot* to the audio frame, which is indicated by means of the *padding_bit* flag [31], [39].

Table 2.3 — *Audio frame duration.*

	f_s 16 kHz	f_s 22.05 kHz	f_s 24 kHz	f_s 32 kHz	f_s 44.1 kHz	f_s 48 kHz
Layer I	24 ms	17.41 ms	16 ms	12 ms	8.71 ms	8 ms
Layer II	72 ms	52.24 ms	48 ms	36 ms	26.12 ms	24 ms
Layer III	36 ms	26.12 ms	24 ms	18 ms	13.06 ms	12 ms

2.3 H.264/MPEG4-Advanced Video Coding

In 2004, the Joint Video Team (JVT) developed an advanced video coding method, known as H.264/MPEG4 Part 10 or H.264/MPEG4-AVC (Advanced Video Coding). This standard proposal combines the coding effort of the ITU-T Video Coding Experts Group (VCEG) with the ISO/IEC JTC1 Moving Picture Experts Group (MPEG). This standard is suitable for cost-effective broadcasting of HDTV and capable of delivering high-quality video at low bit rate, enabling new markets such as handheld television. H.264/MPEG4-AVC operates in a similar manner as previous block-based compression schemes (MPEG-2) and deploys a hybrid-based coding scheme differing on various aspects. The modifications result in a better subjective picture quality and coding efficiency, at the expense of higher complexity. Unlike MPEG-2, which performs prediction at the granularity of a picture resulting in P-frames and B-frames, H.264/MPEG4-AVC performs prediction at slice level resulting in P-slices and B-slices, thereby reducing the granularity of prediction. Furthermore, H.264/MPEG4-AVC separates the video coding from the mapping onto the network layer, where the latter is partitioned into a Video Coding Layer (VLC) and a Network Adapta-

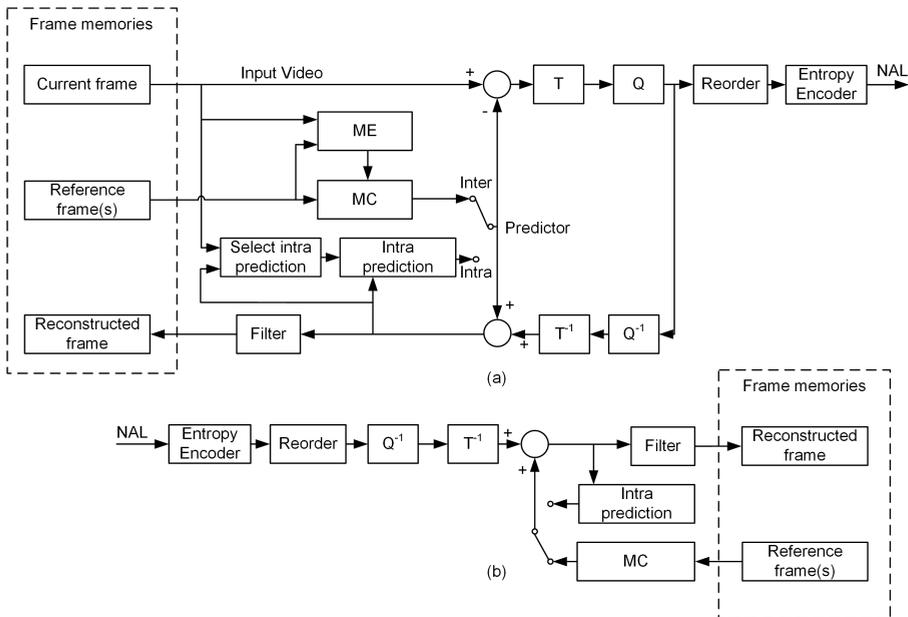


Figure 2.9 — H.264/MPEG4-AVC video codec. (a) Encoder. (b) Decoder.

tion Layer (NAL).

H.264/MPEG4-AVC is based on hybrid coding, involving DCT compression within a picture and motion compensation in the time domain. Figure 2.9 shows a block diagram of an H.264/MPEG4-AVC video codec [40]. Compared to the MPEG-2 coding scheme from Fig. 2.1, both coding schemes show a clear resemblance. However, H.264/MPEG4-AVC has also some clear differences, making this standard far more advanced and at the same time, considerably more complex. Like MPEG-2, H.264/MPEG4-AVC video compression forms a coding toolbox, enabling a wide variety of compression methods, determined by means of *Profiles* and *Levels*. The choice of *Profiles* limits the deployment of the syntax and *Levels* put boundaries on the value of bit-stream parameters [5], [40]. This section presents a few H.264/MPEG4-AVC coding features, which are used in further work in this thesis.

2.3.1 Intra-Macroblock Video Compression

Similar to MPEG-2, H.264/MPEG4-AVC divides the image into a fixed amount of macroblocks, a spatial sub-image of size 16×16 pixels, which is sub-divided into smaller blocks of size 4×4 , or a combination of 4×4 and 8×8 pixel blocks, depending on the deployed profile (Fidelity Range Extension) and the spatial complexity. In H.264/MPEG4-AVC, pixels that construct an intra-coded slice are spatially decorrelated by means of spatial prediction prior to block-based transformation. This differential information is either obtained at macroblock level (16×16 pixels) or at DCT block level of size 4×4 or 8×8 . At the macroblock level, 4 plane prediction modes are available, while for block sizes of 4×4 or 8×8 , 9 prediction modes are available, see Fig. 2.10(b), using adjacent decoded pixels as reference candidates, indicated as gray locations in Fig. 2.10(a,c). Mode 2 is absent from the direction plot, as this mode is based on the average value of available surrounding pixels. For the situation that spatial prediction is performed on an 8×8 DCT block size, the decoded surrounding pixels deployed for spatial prediction are low-pass filtered, prior to construction the 8×8 prediction-block. For the calculation of the 4×4 block-based predictor, the decoded pixels are deployed without low-pass filtering. Intra-coded macroblocks can be deployed by all three slices (I, P and B). After applying spatial prediction, the residual information is transformed involving an integer transform with a size equal to that of the deployed block-based prediction, which is based on 4×4 or 8×8 pixels. Prior to entropy coding, the DC coefficients of the DCT blocks constructing a macroblock are clustered and Hadamard transformed. The coefficients are entropy coded, using either a Context-Adaptive Variable Length Coding (CAVLC), or using an improved method known as Context-Based Adaptive Binary Arithmetic Coding (CABAC) [41].

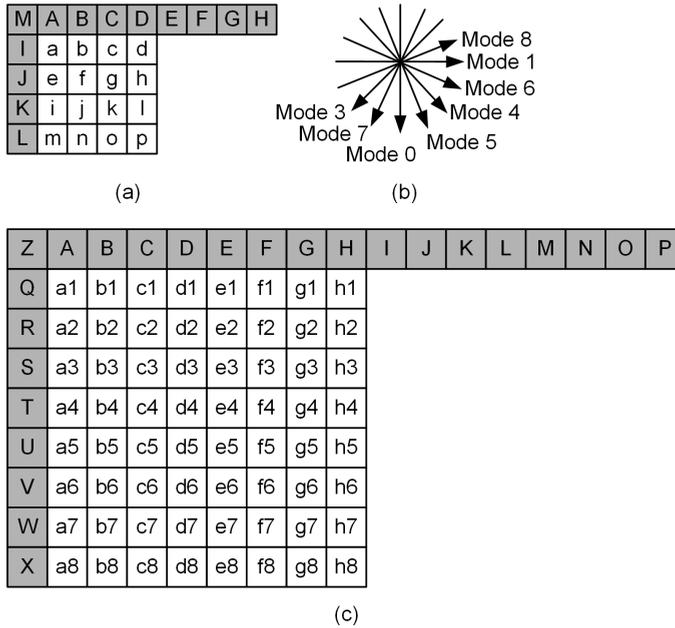


Figure 2.10 — Location of reference pixels for H.264/MPEG4-AVC intraframe spatial prediction. (a) Pixel locations for 4×4 block size. (b) Spatial prediction directions. (c) Pixel locations for 8×8 block size.

2.3.2 Inter-Macroblock Video Compression

Like in MPEG-2, H.264/MPEG4-AVC also supports uni-directional and bi-directional coding, resulting in P-type and B-type slices. Additionally, H.264/MPEG4-AVC supports two more slice types: Switching-P (SP) and Switching-I (SI) providing increased error resistance [41]. A macroblock can be coded in field, frame or field/frame mode, whereby the macroblocks can be further sub-divided into smaller blocks, each equipped with their own motion vectors. The motion vectors are calculated at a quarter-pixel accuracy, resulting in up to 16 motion vectors per macroblock. Unlike MPEG-2, in H.264/MPEG4-AVC a macroblock can be predicted from multiple reference pictures, leading to additional reference information for each 16×16 , 16×8 or 8×16 macroblock partition or 8×8 sub-macroblock, indicating from which picture the prediction is obtained. H.264/MPEG4-AVC supports also skipped macroblocks. However, unlike in MPEG-2, the motion properties differ in the sense that they are not assumed to be zero, but derived from neighboring macroblocks. Another difference between MPEG-2 and H.264/MPEG4-AVC is that P-slices can

be predicted from multiple decoded pictures, whereas B-slices can be used as a predictor. The difference between P-type slices and B-type slices is that a macroblock in B-type slices may be constructed on the basis of a weighted average of two motion-compensated predictions. The predictors are obtained from arbitrary reference pictures. The indexing is arranged according to two picture lists known as List 0 and List 1. Three different types of inter-picture predictions are distinguished in B-slices List 0, List 1 and bi-predictive, depending on from which reference picture list the prediction originates. Macroblocks or sub-macroblocks constructing a B-type slice, may be directly coded, by omitting associated motion data. If no residual information is left, this results in skipped macroblocks, which are efficiently coded.

2.4 Video Coding Artifacts

Cost-effective video broadcasting is obtained by removing irrelevant information and attenuating some relevant video information. Irrelevant video information contains e.g. high-frequency video information that cannot be observed under certain conditions, whereas relevant video information is mostly the remaining low- and medium-frequency video information. When video coding is operated at a cost-effective bit rate, then typical coding artifacts are introduced, such as blocking, ringing and mosquito noise [35], [42].

- Block visibility artifacts are introduced due to block-based coding scheme, featuring independent quantization of blocks. This quantization influences information crossing block borders, giving rise to discontinuities in signal waveforms, which results in visible vertical or horizontal edges at the DCT-block borders, see Fig. 2.11(a).
- Ringing is an artifact of a more fundamental nature, caused by the truncation of frequency components constructing waveforms (called Gibbs phenomenon). The visible appearance depends on the deployed block size, the amount of quantization and the waveform positioning in the block. As a result, ringing visibility occurs mostly when an edge, either horizontal or vertical, crosses the whole height or width of a transform block, see Fig. 2.11(b).
- Mosquito noise is a coding artifact that originates from either the truncation or removal of high-frequency information, or the prediction mismatch of the Motion-Compensated (MC) prediction. The distortion has typically low intensity and can be both static and dynamic. The degradation is clearly visible in low-frequency regions [43], see Fig. 2.11(d), especially when the video signal is enhanced by applying sharpness enhancement.

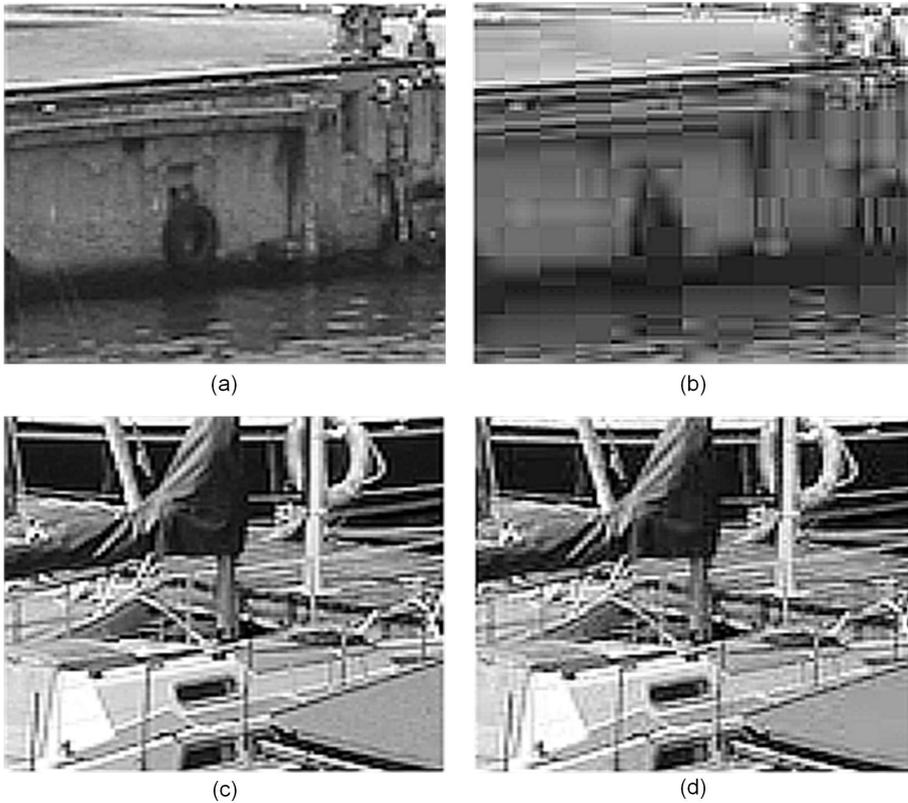


Figure 2.11 — Example MPEG-2 coding artifacts (zoom factor is two). (a) Original image fragment. (b) Blocking and ringing artifacts. (c) Original image fragment. (d) Mosquito artifacts.

For all three distortions, the block size of the deployed transformation plays a crucial role. MPEG-2-based video compression employs an 8×8 DCT transformation. Due to this relatively large block size, artifacts such as ringing and mosquito noise can deteriorate a substantial amount of pixels in the reconstructed block. H.264/MPEG4-AVC has addressed the visibility aspect of artifact appearance, by employing a smaller 4×4 pixel block. However, the Fidelity Range Extension (FRExt) Profiles allow to use an 8×8 -based transformation, thereby potentially ruining this benefit. As a result, with a 4×4 pixel block, the number of pixels that can be potentially degraded is less, masking ringing and limiting the visibility of mosquito noise. Blockiness, i.e. artifacts that are located at the block boundaries, is deteriorating the perceived video

quality. H.264/MPEG4-AVC has therefore an adaptive in-loop de-blocking filter reducing the blockiness, which limits the diffusion of blockiness into other pixel blocks due to the MC prediction process.

In order to improve the picture quality, video post-processing is regularly applied by the Consumer Electronics (CE) industry. Typically, this processing involves locally-adaptive low-pass filtering, which attenuates the detected coding artifacts, thereby avoiding excessive image blur. Dynamic artifacts, e.g. mosquito noise due to Motion Compensation (MC) mismatch, are well suppressed when applying Temporal Noise-Reduction (TNR) or Motion-Compensated Temporal Noise-Reduction (MCTNR) [44], [45]. However, for the situation that the mosquito noise is static, a spatial filtering approach is required. Detection of mosquito noise regions typically involves a two-step approach, as there is not a single metric revealing potential degraded locations and allowing the distinction between intended texture or artifact contamination. As a majority of the papers on mosquito noise indicate that this noise appears near the border of objects, the first step typically determines the locations of edges, from which the degraded locations are derived. On the basis of the detected locations, an adaptive low-pass filter attenuates the artifact degradation.

2.5 Personal Video Recording

The availability of high-capacity hard-disc drives has fueled the development of Personal Video Recording (PVR). Such systems are consumer-based storage products for audiovisual information. These modern storage systems are equipped with rich features that go beyond the possibilities of traditional tape-based storage solutions. Figure 2.12 summarizes the main PVR features for which solutions have been developed in the past decade. Basically, a PVR can be network-based (client-based) [46], or operate in a standalone fashion. In the former situation, storage is performed by the service provider, which facilitates various features such as enhanced playback. For the latter case, the PVR operates autonomously at the client side, where advanced features need to be established based on locally derived information [47]–[50]. The combination of network-based PVR and client-based PVR is a hybrid PVR. This concept enables special configurations for PVR usage, such as the PVR operating as a server in a network, which is accessible by authorized remote devices like a smartphone for local playback [51]–[54]. A basic PVR features conventional trick play, such as fast search and slow motion, and has time-shift recording [55]. Alternatively, a more advanced PVR has features such as semi-automated video editing [56], video transcoding [57] and advanced forms of video navigation for inter- as well as intra-program video navigation. For stored audiovisual programs, video navigation can be separated into two categories:

video browsing and video retrieval, see Fig. 2.13. Advanced video *browsing*, also known as intra-program navigation, has the objective to increase the navigation efficiency through a recorded program [52], [58]–[60]. Another form of advanced video navigation is inter-program video navigation, enabling personalized program guides for guidance through available broadcast channels and personal program lists for navigation through stored programs, thereby increasing the search efficiency of broadcasted and recorded programs [61]–[64]. Video navigation has been subject of research for many years with the objective to improve the efficiency, exploiting methods to condense the spatial-temporal information and the presentation of the condensed information. Improvements regarding video navigation have resulted in e.g. text-based browsing [65], key-frame extraction [66] and program summarization [67], [68], but also in attractive rendering of the video navigation information [69]–[71].

Advanced intra-program video navigation provides solutions for reducing the stored spatial-temporal program information and the associated rendering of that reduced spatial-temporal information. The spatial-temporal information can be decreased by interpreting program information such as closed caption text [72], [73], and audiovisual information like character recognition [74], speech recognition [75] or scene-change detection [76]. These detections, generate time markers associated with a certain characteristic on the program-time axis. These time markers are typically stored as Characteristic Point Informa-

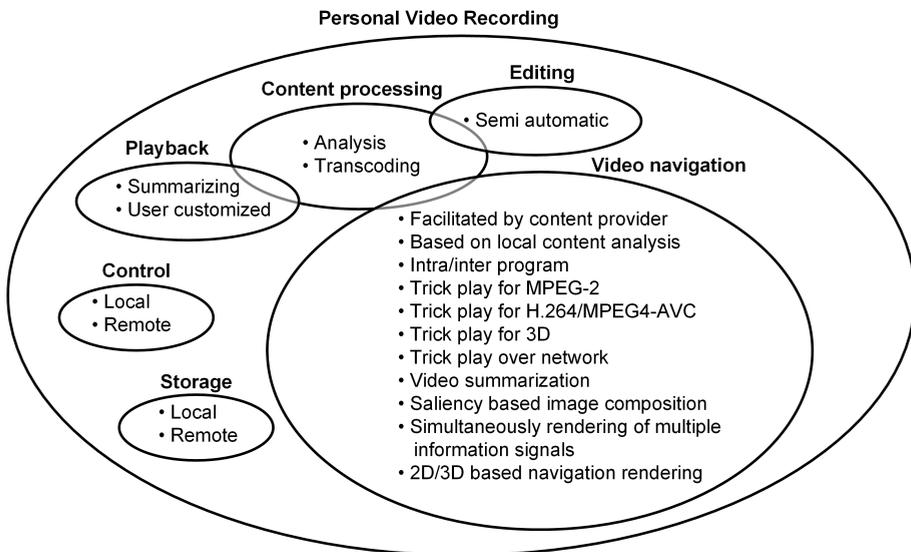


Figure 2.12 — Overview of PVR features.

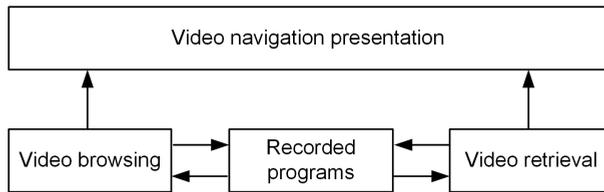


Figure 2.13 — *Stages in PVR video navigation.*

tion (CPI), which is basically a metadata information signal, associated with a particular recording [73], [77], [78]. Besides the storage of time markers, enabling efficient browsing through a stored program using various methods e.g. time-adaptive skip forward or skip backward [79], CPI may e.g. also contain thumbnail-sized images, which are representative for a particular scene or may store string-based information enabling text search [65], [80].

The availability of thumbnails enables video browsing, deploying a navigation bar/column or navigation plane [70], [81]–[83]. Thumbnail-based navigation may be further enhanced by additional information, such as text, presenting instantaneously information from multiple information sources, enhancing the navigation efficiency [72], [73].

With the appearance of 3D graphics more attractive presentation methods are proposed, utilizing objects shapes such as e.g. a rotating ball, carrousel, or helix on which the video navigation information is depicted [71].

The usage of audio information for trick play has been limited to either playback speeds around unity [14], or for feature extraction enabling advanced intra-program video navigation [75].

2.5.1 Intra-program video navigation

Intra-program video navigation, also known as video browsing or trick play, is defined as video playback in non-consecutive or non-chronological order, as compared to the original capturing order. Trick play can be divided into conventional fast-forward/reverse playback or slow-motion forward/reverse playback, and advanced search methods, referring to modern forms of trick play. The conventional forms are found in analog and digital tape-based video recorders. The latter has become possible for storage systems deploying random-access media, such as disc and silicon-based memories. This section covers the basic aspects of conventional trick play and briefly elaborates on low-cost trick play deployed for pull-based and push-based storage system architectures. The navigation methods are presented without implementation aspects.

A. Conventional video navigation

For conventional trick play, a distinction is made between fast-search and slow-search mode. Let P_s be the relative playback speed, which is unity for normal play, then fast-search trick play is obtained for $|P_s| > 1$, and slow-motion trick play for $|P_s| < 1$. Although this is a firm separation between fast search and slow motion, there is an overlapping area for playback speeds in the vicinity $L_b < P_s < U_b$, with $L_b < 1$ and $U_b > 1$ of the unity playback speed.

Fast-search video navigation is characterized by that the pictures forming the trick-play sequence are derived from the normal-play sequence by applying a temporal equidistant subsampling factor, which corresponds to the intended playback speed P_s . In practice, P_s is limited but not restricted to integer values, see Table 2.4.

Slow-motion search is obtained by repetitive display of each normal-play picture. The amount of repetitions is equal to the reciprocal of P_s . Again, practical values for P_s are limited but not restricted to integer values. Although the basic operation to obtain slow motion has low costs, a distinction is made between slow motion on progressive video and interlaced video. When video originates from an interlaced video source, repetition of an interlaced picture may result in motion judder. Such a situation occurs when the spatial area contains an object that is subject to motion between the capture time of the odd and even field, forming a single frame picture. Repetition of such an interlaced picture causes the object repetitively traveling along the trajectory, which is perceived by the viewer as motion judder [12].

The above solutions for trick play are based solely on using resampled video information. For playback speeds in the vicinity of unity, there is an alternative implementation for trick play, that builds on also re-using the normal play

Table 2.4 — *Examples of playback speeds for conventional trick-play modes.*

Playback mode	Typical playback speed P_s	Information signal
Fast search	4, 8, 12	video
Slow motion	$1/2, 1/3, 1/4$	video
Pitch control	$4/5, 3/2,$	video/audio

audio information. For this situation, time-scaled normal-play audio information is used to create an audiovisual trick-play sequence. To maintain audibility of the time-scaled audio information, pitch control is required. The playback speeds P_s that can be used for this type of trick play depends heavily on aspects of the normal-play sequence, such as speed of the oral information and the algorithm used for processing the audio signal. Algorithms for time- and pitch-scaling can be divided into frequency-domain and time-domain methods. Good results are obtained using Pointer Interval Controlled OverLap and Add (PICOLA) [84], an algorithm that operates in the time-domain on mono-channel audio signals for playback speeds in the range $0.8 \leq P_s \leq 1.5$ for audiovisual content with spoken text. This form of trick play indicates that P_s does not need to be limited to integer-based values.

B. Efficient video navigation

For fast-search trick play there are navigation-efficiency limitations caused by the final trick-play video-sequence quality. The root cause for this video-quality penalty is either caused by the trick-play information retrieval or imposed by the video refresh-rate in combination with the applied trick-play playback speed. Section 2.5.1 - C briefly discusses the video-quality penalty caused by the trick-play information retrieval process, while the video quality declines due to the combination video refresh-rate and playback speed is discussed in this subsection. The discussion on video-quality degradation, based on a combination of trick-play playback speed and video refresh-rate, requires the definition of navigation efficiency.

We define navigation efficiency, as the ability to successfully interpret the content of the received video information by the viewer, and consider that information to be suitable for navigation, while searching the normal-play video sequence. In order to obtain insight on the meaning of this definition, we discuss the following mindset experiment.

Let us assume a normal-play video sequence constructed from non-correlated scenes, where each scene has a 3-second duration. Conducting conventional fast-search trick play on this experimental scene, at maximum playback speed, would require that $P_s = 75$ for 25-Hz television signals. In this case, each normal-play scene interval of 3-seconds provides a single picture, that contributes to the fast-search trick-play sequence. This trick-play video sequence has been tested by a test panel to solve the following question: "Does the conventional trick-play navigation efficiency decline for high playback-speed?"

With 69 % the test panel confirmed that the navigation efficiency declines for a high playback-speed, whereas 8 % answered with *No* indicating no navigation efficiency decline, while 23 % answered with *Don't know*, neither confirming nor denying a navigation-efficiency issue. Although in the above example each scene of the normal-play video sequence contributes to the fast-search

video sequence, 69 % of the test panel considers the navigation efficiency to be negatively affected. The root cause for this navigation-efficiency drop is the lack of correlation between successive pictures, forming the trick-play video sequence. As a result of this, multiple images of a single scenery are required, in order to be successfully interpreted by a viewer. Experiments reveal that for a typical normal-play video sequence, a good visual performance is achieved when a trick-play video sequence is constructed using 3 correlated pictures, which results in a practical upper bound of $P_s = 25$ for the fast-search trick-play speed.

C. Trick play for digital storage system

With the introduction of MPEG-based video compression, digital storage systems came on the market with features comparable to features of analog tape-based storage systems. For tape-based storage systems, trick play is a feature that is realized by means of the servo system, controlling the tape speed in relation to the scanner's scan-path. Trick-play on a tape-based analog recording is obtained on the basis of re-used normal-play field-based video fragments. Hereby, during trick-play playback, a noise bar appears between each retrieved fragment, influencing the final video navigation quality. The number of noise bars increase, when increasing the trick-play playback speed. In digital tape-based storage systems, besides servo control employing either speed-lock or phase-lock based playback, see Fig. 2.14(d)(e), also the information stored on the locations read by the scanner plays an essential role for fast-search trick play. As a result, the locations read by the scanner during trick-play playback, should/may be filled during record with the final video trick-play data, see Fig. 2.14. In the previous text, we purposely used 'should/may' to distinguish two system approaches. In a first approach, specific locations on tape are allocated to contain dedicated trick-play data, which corresponds to a particular playback speed. This provides a guaranteed trick-play performance for a particular playback speed, but requires additional storage capacity on tape. In the second approach, data is distributed over the tape format, and is made accessible in small segments, enabling trick-play picture refresh for certain playback speeds. In this way, additional overhead is circumvented, while providing region-based picture refresh, which is beneficial for low-speed trick play. For optical disc-based video-storage systems, trick play has been a basic feature, providing fast-search and slow-motion trick play, enabled by the associated standard and facilitated by the random access of the storage medium. Basically, fast-search trick play is realized on the basis of re-used intra-coded normal play video information. In order to efficiently locate these intra-coded

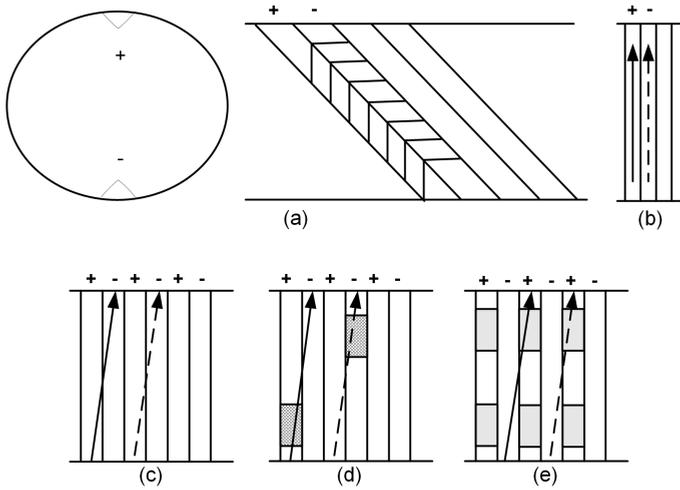


Figure 2.14 — Trick play for digital tape-based consumer storage systems deploying helical-scan recording. (a) Two-head scanner, with different azimuth. (b) Scanner scan-path for normal-play playback. (c) Scanner scan-path for playback speed twice the normal speed (fast-search trick play). (d) Locations on tape which can be read by the head with correct azimuth, servo deploys phase-lock. (e) Locations on tape that can be read by the head with correct azimuth, servo deploys speed-lock.

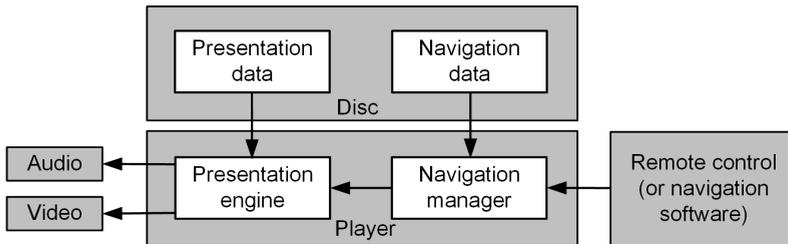
pictures, early standards such as Video Compact Disc (VCD) or Super Video Compact Disc (SVCD), as well as modern standards such as Digital Versatile Disc (DVD) or Blu-ray Disc (BD), incorporate mechanisms revealing the location of intra-coded pictures on the optical disc [14]. In this way, a low-cost video navigation solution can be cost-effectively implemented. These standards assume a playback speed equal to unity. For an optical disc-drive that supports higher playback speeds, other forms of fast-search trick play can be realized.

D. Pull-based versus push-based trick-play video decoding

The storage-system architecture influences the involved trick-play signal processing. A digital storage system can have either a pull- or a push-based architecture. In a pull-based architecture, the video decoder pulls the data from the storage medium, whereas in a push-based architecture, the video decoder receives the audiovisual information compliant to the underlying standards. For the pull-based architecture, constraints such as bit rate and buffer sizes imposed by the standards are not relevant, as the decoder pulls the data of the record medium. For the push-based architecture, constraints imposed by the

(mm:ss:ff)		
00:00:00	Pre-gap	150 sectors
00:02:00	User Area	-
00:02:16	Primary Volume Descriptor	ISO 9660
	-	
00:04:00	SuperVCD Information files	Disc Information "INFO.SVD" (mandatory) Entry table "ENTRIES.SVD" (mandatory) List ID Offset table "LOT.SVD" (extension) Play Sequence Descriptor "PSD.SVD" (extension) Search table "SEARCH.DAT" Tracks table "TRACKS.SVD" (mandatory)
	Segment Play Item area	Segment Play Items
	Other Files	"EXT" directory "SCANDATA.DAT"

(a)



(b)

Figure 2.15 — *Trick play facilitated by the optical storage standard. (a) Navigation facilitated for SVCD. (b) Navigation facilitated for DVD.*

applied standards need to be satisfied, in order not to jeopardize correct playback. For traditional trick play, i.e fast search and slow motion, there is a difference in performance due to the different nature of pull- and push-based architectures and variations in the deployed trick-play signal-processing concepts. For a pull-based architecture, the performance depends on the data retrieval rate, processing speed of the multimedia decoder and the presence or absence of random access. For a push-based architecture, the performance is influenced by the constraints of the underlying standards. For audiovisual storage systems based on a push-based architecture using the MPEG-2 standard [5], trick play can either employ the trick-play signaling information provided by the Packetized Elementary Stream (PES) header, see Section 2.2.2, or generate an MPEG-2 trick-play compliant stream without the trick-play signaling informa-

tion [12]. The trick-play signaling information provided by the MPEG-2 PES header is used to control the output of the video decoder during trick play. Although this solution is specified by MPEG-2, its support is optional and not obligatory [37].

E. Influence of normal-play video encoding parameters on trick-play video

Modern compression schemes such as MPEG-2 or H.264/MPEG4-AVC, achieve high compression ratios by exploiting spatial and temporal correlation in the video signal. Compression of pictures exploiting only spatial information are intraframe-coded, whereas compression of pictures having temporal correlation are interframe coded, leading to P- and B-type pictures for MPEG-2 and P- and B-type slices for H.264/MPEG4-AVC compressed video. In a compressed video sequence, the distance between two successive intraframe-coded pictures is expressed by N , which is also known as the Group-Of-Pictures (GOP) length, whereas the distance between P-type predictive pictures is expressed by M . For the situation that $M > 1$, the number of B-type pictures preceding a P-picture is equal to $M - 1$. In general, trick play for digital consumer storage equipment is a low-cost feature, which limits the amount of involved signal processing. Low-cost trick-play algorithms deploy pictures that are selected from the compressed normal-play sequence. For fast-search trick play, the minimum fast-forward search speed is equal to M , whereas other fast-forward speeds are obtained for speed-up factors equal to N , or typically an integer multiple of N ³. Furthermore, note that there is basically a gap between speed-up factor M and N if $N \gg M$. This gap is caused by the fact that P-type pictures can only be decoded if the reference (anchor) picture has been decoded. For H.264/MPEG4-AVC this situation is even worse, as P-slices may refer also to B-slices, which leaves only I-slices to be deployed for trick play. For typical video compression applications, such as Digital Video Broadcast (DVB) or digital recording, the GOP size $N = 12$ and P-picture distance $M = 3$ results in fast-search speed-up factors $P_s = \{3, 12, 24, \dots, 12n\}$, with $n = \{1, 2, 3, \dots\}$. Fast-search trick play in reverse direction is obtained in a similar way, but for a speed-up factor equal to M , buffering is required to store the decompressed pictures to facilitate reordering. This is required to match (potential) motion with the reverse playback direction, as the decoding is always performed in the positive time direction. This forward decoding direction introduces an extra delay, which occurs only once when switching to the reverse-search mode with speed-up factor equal to M .

³Note that additional fast-search speed-up factors are possible, where the trick-play speed is an integer divider of the GOP size N . This leads to an implementation, where the I-picture is repeatedly displayed until the next I-picture can be decoded, where the repetition rate is related to the selected trick-play playback speed.

MPEG-2-compliant video navigation

3.1 Introduction

Video navigation is an essential Personal Video Recording (PVR) feature, which encompasses basic navigation methods like fast-search and slow-motion, as presented in Section 2.5.1. However, conventional fast-search video navigation, typically indicated as trick play, is solely controlled by the playback speed P_s , which limits the usability. This is caused by the fact that the navigation efficiency limitation from Section 2.5.1 in terms of rendering time, does not scale well for a broad range of trick-play playback speeds, as discussed in 2.5.1. It was argued that the maximum video navigation playback speed is limited to a practical value of around 25 times the normal-play speed.

In the past decade, a broad range of features for PVR have been proposed, either improving existing or adding new functionality, as discussed in Section 2.5. In this chapter, three video navigation use cases are proposed, each addressing a particular video navigation time interval, over which the navigation is conducted. For each use case, one or more efficiency aspects, as depicted in Fig. 3.2, are considered. We have decided to adopt the following features.

- *Interoperability.* In general, interoperable networked communication systems employ communication standards that enable smooth and seamless communication. Communication systems employing international standards invoking non-mandatory communication protocols of that standard, may cause an undefined behavior at the receiver side, e.g. when that receiver is based on mandatory parts only. For this reason, client-server communication should be based on supported (mandatory) communication protocols, in order to avoid such undefined receiver behavior.
- *Network-based playback of video navigation information.* Network-based storage systems rely on interoperability, enabling connectivity between heterogeneous networked systems for all PVR playback modes. This implies that the communicated information adheres to (MPEG) standards,

of which the communication should be supported by client-server systems.

- *Navigation-plane-based video navigation.* Hereby, the usage of a navigation plane, constructed in the form of a mosaic screen, alleviates the refresh-rate constraint from the individual picture rendering process. This solves the problem of perceived information updating caused by high-speed conventional trick play.
- *Rendering of multiple information sources.* Employment of normal-play audiovisual information, enables full exploration of the human perception, which employs both visual and auditory cues.

Intra-program video navigation is a function for which different solutions are required to control the perceived information. For this reason, we divide intra-program video navigation into three use cases, each addressing a particular time interval over which the video navigation is conducted.

Besides interoperability aspects and navigation efficiency, there are several essential system aspects such as latency, Quality-of-Service (QoS) and complexity that influence a possible video navigation solution. Any video navigation solution should in the end be deployed on low-cost DTV platforms, thus requiring cost efficiency.

The sequel of this chapter is as follows. First, Section 3.2 introduces the three video navigation use cases, each addressing a specific video navigation time interval. Section 3.3 presents two conceptual solutions enabling short-time interval video navigation in a network environment. Section 3.4 elaborates on the proposed short-time interval video navigation solution. In Section 3.5, we present two conceptual solutions for solving long-time interval video navigation. The proposed long-time interval video navigation solution is presented in Section 3.6, while Section 3.7, discusses implementation and experimental results. Chapter 3.8 presents a short discussion on future options for the proposed mosaic screen navigation concept. This chapter finishes with Section 3.9 presenting the conclusions.

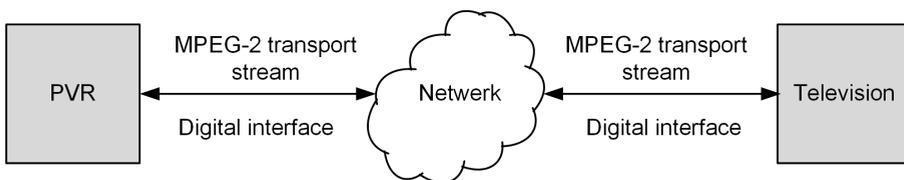


Figure 3.1 — *Networked personal video recording.*

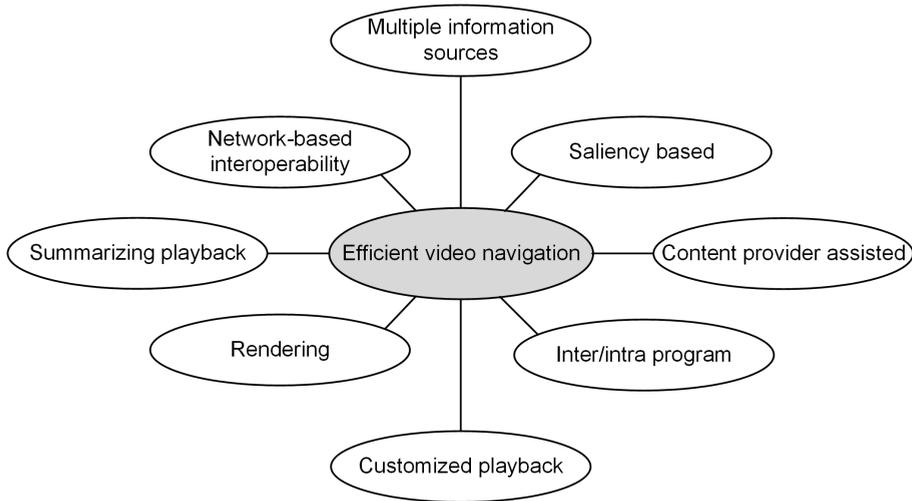


Figure 3.2 — *Main efficiency axis for video navigation.*

3.2 Proposed video navigation use cases

In the past two decades, digital video has become available for consumer applications [3], [85]–[87]. A transition from analog to digital video recorders has been made via tape-based systems, resulting in storage systems based on media, such as hard disk (HDD), Solid State Disc (SSD) and optical disc (DVD/BD). Due to the continuous growth in storage capacity, the need for efficient navigation such as fast search is augmenting. Furthermore, due to the random access of non-tape-based storage systems and an increase in signal processing capabilities of employed platforms, new navigation methods are possible (see Fig. 3.2), contributing to the intra-program video navigation efficiency.

Intra-program video navigation is challenged by the increased storage capacity of digital storage systems, where efficiently locating a specific scene can become a daunting task. Video navigation, which can either be autonomous or interactive, is further challenged when a storage system is operated in a client-based manner, which requires video navigation information to be transmitted across a digital interface, see Fig. 3.1. Efficient video navigation has been subjected to research for many years, providing efficient solutions in various directions, as indicated by Fig. 3.2. Typical PVR platforms are constrained on key system resources such as available memory bandwidth, cycle budget and memory. Since navigation is an additional feature to the storage system, video navigation solutions benefit from employing techniques, such as computational

complexity scalability [88] and/or signal processing in the MPEG-compressed video domain. Furthermore, computation-intensive operations are preferably shifted to the MPEG video decoder, contributing to a complexity reduction at the server side.

We have defined three video navigation use cases, each addressing a particular video navigation situation, based on the time interval over which the video navigation is conducted. Depending on the video navigation use case, a cost-effective autonomous or interactive solution is proposed, enabling local and/or client-based video navigation, employing selected efficiency aspects: (1) network-based video navigation, (2) multiple information sources and (3) plane-based rendering.

- **Use case 1.** *Autonomous client-based video browsing over a short-time interval.* Autonomous video browsing over a short-time interval, having a typical duration of up to a few minutes, is a frequently deployed feature during normal play, which is conducted either in forward or reverse direction. This type of video browsing leads to subsampling or repetition of pictures, which are derived from the normal-play video sequence. The former processing type is known as fast-search trick play, whereas the latter is known as slow-motion trick play. In a networked home, the storage location may differ from the decoding location, resulting in client-server-based playback. When deploying MPEG-compliant trick play, interoperability is guaranteed, thereby simplifying the networked operation. This type of video navigation is further elaborated in Section 3.4.
- **Use case 2.** *Interactive client-server-based video browsing for a long-time interval.* The video navigation efficiency of conventional fast-search trick play declines when used at high playback speed, as pictures constructing a fast-search trick-play sequence have a reduced correlation when increasing the trick-play playback speed, as discussed in 2.5.1. The navigation efficiency is improved when decoupling the navigation from the refresh-rate of the individual picture rendering process. To this end, we propose an MPEG-2-compliant hierarchical mosaic-screen navigation method. By employing different temporal subsampling factors for the individual hierarchical layers, a zoom-in or zoom-out operation on the temporal video information is facilitated, enabling efficient browsing through a large amount of video information. This type of video navigation is further elaborated in Section 3.6.
- **Use case 3.** *Autonomous audio-enhanced video browsing for a medium-time interval.* Video browsing over a medium-time interval, having a typical duration up to 30 minutes and conducted either in forward or reverse direction. This is a feature which is typically deployed when searching for a particular scene, or to quickly obtain a global impression of the

stored audiovisual content. As human perception employs both visual and auditory cues, it is opportune to employ audio information associated to the video-navigation information. When rendering detailed audiovisual normal-play information fragments, the viewer may lose the overall fast-search experience. This fast-search experience is maintained when simultaneously rendering a fast-search navigation signal in combination with normal-play audiovisual fragment information. In this way, the video browsing method can also be deployed as a viewing mode, while the simultaneous rendering of multiple information sources, enhances the video navigation efficiency. Hereby the trick-play information signal provides a coarse overview, whereas the normal-play fragments provide detailed information. This type of video navigation is further elaborated in Chapter 4.

3.3 Conceptual MPEG-2-compliant video navigation

In this section, we provide a conceptual outline for solving MPEG-2-compliant fast-search and slow-motion video navigation over a network, which can both be realized in two ways. Both solutions follow the concept for fast-search and slow-motion trick play, based on temporal resampling the normal-play video sequence, as introduced in Section 2.5.1. Hereby, the first solution conducts video transcoding, while the second solution is based on re-use of MPEG-2-compressed normal-play video information. For both solutions, a conceptual architecture is presented including performance estimates on the used key system resources, revealing the complexity of both solutions.

A. First concept: Transcoding of MPEG-2 video data

In a first approach, a trick-play video sequence representing either fast-search or slow-motion, is derived from an MPEG-2-compressed normal-play video sequence invoking: MPEG-2 decoding, temporal resampling and encoding (transcoding) of the selected normal-play video information, see Fig. 3.3(a). At the left side of Fig. 3.3(a), MPEG-compressed data is retrieved from the storage medium and passed on to an MPEG-2 demux to access the video Elementary Stream (ES), consisting of the individually encoded pictures. In order to apply trick play on such a sequence, the video ES is fully decoded, resulting in decompressed pictures. For fast-search trick play, as discussed in Section 2.5.1, playback is obtained via temporal subsampling of the original sequence of normal-play pictures. Alternatively, for slow-motion playback, individual pictures from the original normal-play picture sequence, are repeated to slow down the actual playback. For these two playback modes, the newly derived sequences of pictures are again MPEG encoded and packetized, forming

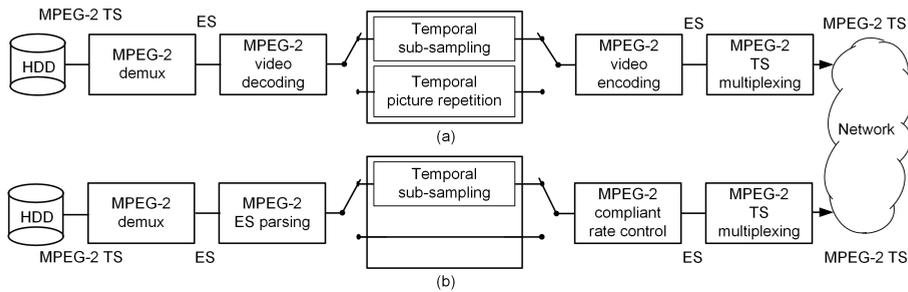


Figure 3.3 — *MPEG-2-compliant trick-play video processing. (a) Trick play based on transcoding. (b) MPEG-2-compliant trick play based on re-used normal-play compressed pictures.*

a new MPEG-compliant stream, which can be transmitted across a network to any type of client. The previously described process requires full decoding and re-encoding, which explains the term transcoding.

Employing this concept for deriving a fast-search video navigation signal, results in an estimated load on key system resources, as depicted in Table 3.1. The figures in Table 3.1 for fast-search apply to full decoding of a 25-Hz Standard Definition (SD) video signal with a 4:2:0 color format and intraframe encoding of the fast-search video trick-play sequence. Table 3.1 also indicates performance estimates for deriving a slow-motion video navigation sequence, whereby the slow-motion information is MPEG-2 encoded using MP@ML.

Normal-play decoding of an MPEG-2 MP@ML encoded video sequence, involves the retrieval of information from one or two reference pictures, resulting in a doubling or tripling of the full bit rate of a single uncompressed video sequence. For a 25-Hz SD video sequence with 720 pixels and 576 lines and 4:2:0 sampling format, the uncompressed video bit rate equals 124.42 Mbit/s and involves a memory capacity requirement of 608 kBytes per full-color frame ($720 \times 576 \times 1.5$ pixels).

When deriving a fast-search video navigation sequence on the basis of full decoding, the MPEG-2 decoding process basically scales with the applied trick-play playback speed. For example, when applying a fourfold search, the retrieved normal-play video sequence has to be decoded at four times the normal-play speed, in order to access the corresponding pictures constructing the fast-search trick-play sequence. This is a costly solution, requiring considerable system resources, in particular for high playback speeds, where the processing quickly becomes too expensive for a consumer device. The quickly growing processing costs are hampered by the lack of parallel decoding, which is not possible due to a strong time-dependence of the individually encoded pictu-

Table 3.1 — *Estimated system-resource utilization for full decoding of MPEG-2-compressed MP@ML normal-play SD video and corresponding MPEG-2 encoding.*

Speed-up factor	System resources			
	Decoder Mb/s	Encoder Mb/s	Memory capacity kBytes	DSP/CPU cycle load
2	776	139	2,430	High
4	1,552	139	2,430	Very high
11	4,270	139	2,430	Extreme high
1/2	194	388	3,645	High
1/3	130	388	3,645	High
1/5	78	388	3,645	High

res. This load on system resources is further increased when augmenting the normal-play SD video resolution to High Definition (HD) or even Ultra High Definition Television (UHDTV/UltraHD). For example, the throughput rate of HD signals increases with a factor five.

Let us proceed with an example how the throughput rate increases with a higher playback speed, see Table 3.1. Using the MPEG-2 decoder from Fig. 2.1(b), the consumed bandwidth involves 15 Mbit/s video ES and three times the 124.42 Mbit/s for two motion-compensated pictures and one output stream, resulting in a total decoder throughput of 387.25 Mbit/s. For the situation that the playback speed $P_s = 2$, the throughput rate for MPEG-2 decoding is doubled, resulting in 776.5 Mbit/s.¹

The generation of a slow-motion video navigation signal differs from fast search in the sense that the decoding load declines, when the slow-motion playback speed also decreases, while the encoding load remains constantly high, as depicted in Table 3.1. The load on system resources, for both fast-search and slow-motion navigation methods, is further increased considering the involved MPEG-2 demultiplexing and multiplexing of the compressed video ES into a compliant MPEG-2 Transport Stream (TS).

B. Second concept: Re-use of MPEG-compressed normal-play video data

In a second approach, a video navigation sequence, either fast search or slow motion, is based on re-use of normal-play MPEG-2-compressed video information, avoiding video transcoding prior to transmission, see Fig. 3.3(b). At the left side in this figure, MPEG-compressed data is retrieved from the storage

¹For simplicity, the values in Table 3.1 have been truncated.

medium and passes through an MPEG demux to access the video ES, consisting of the individually encoded pictures. For trick play, as discussed in Section 2.5.1, fast-search playback is obtained via temporal subsampling of the MPEG-2-compressed video ES containing the sequence of normal-play pictures. Alternatively, for slow-motion playback, individually MPEG-2-compressed pictures from the video ES are repeated to slow down the actual playback. For these two playback modes, the newly derived sequence of MPEG-2-compressed pictures in ES format, is forwarded to the final MPEG-2 multiplexer, after being re-formatted into a final video navigation ES by the MPEG-2 rate control block. This functional block fulfills a dual rate control: (1) bit-rate control of the final video navigation ES via insertion of locally generated repetition pictures, (2) frame-rate control also via insertion of the same type of repetition pictures. The concept of repetition pictures will be further elaborated in Section 3.4. This second concept only re-uses MPEG-2-compressed ES pictures, which leads to: (1) the absence of transcoding and (2) the re-use of only compressed pictures, explaining the term “re-use” to indicate the algorithm. A further advantage of this concept is that the final generated video navigation signal is MPEG-2 compliant. This circumvents the usage of the trick-play signaling information, provided by the MPEG-2 PES header, to control the output of the video decoder during trick play. Although this solution is specified by MPEG-2, its support is optional and not obligatory as discussed in Section 2.5.1.

Let us now address the effect on involved system resources of this second video navigation concept aiming at re-use. The main involved signal processing is reduced to bit-stream parsing, thereby decoding picture headers and selecting MPEG-2-compressed pictures, which form the video navigation signal. Table 3.2 indicates estimated load on required key system resources for parsing and buffering MPEG-2-encoded pictures (MP@ML) forming the video navigation signal. Such a trick-play stream contains intra-coded pictures and optionally, locally derived P-compressed pictures. The primary objective of the locally generated P-pictures is to conduct rate control. The algorithm details for this rate control are further elaborated in Section 3.4.

Let us now further detail the required throughput rate and memory occupation of this video navigation approach. The maximum input bit rate of the MPEG-2 encoded normal-play video ES is determined by the employed MPEG-2 *Level*, which is 15 Mbit/s for the *Main Level*. When conducting fast search on such a video ES, the throughput rate scales with the applied fast-search playback speed. For example, when applying a fourfold search, the throughput rate is increased to 60 Mbit/s. The potential size of an I-picture equals the input buffer size of an MPEG-2 decoder, which equals 224 kBytes. When applying double buffering, the total memory capacity becomes 448 kBytes, as depicted in Table 3.2. The load for demultiplexing, parsing, extracting, separately buffering of re-used pictures and final multiplexing results in an acceptable system resource load, while the cycle load on a DSP/CPU for the involved signal pro-

Table 3.2 — *Estimated system-resource utilization for full re-use of MPEG-2-compressed MP@ML normal-play SD video.*

Speed-up factor	System resources			
	Stream parsing Mb/s	Stream transmission Mb/s	Memory capacity kBytes	DSP/CPU cycle load
2	30	15	448	Modest
4	60	15	448	Modest
11	165	15	448	High
1/2	8	15	448	Modest
1/3	5	15	448	Modest
1/5	3	15	448	Modest

cessing is expected to be quite modest typically.

We have discussed two approaches for deriving a video navigation signal and estimated their utilization of system resources. A comparison of the two approaches on the basis of the utilized system resources, clearly reveals that the second navigation processing concept based on re-used compressed pictures is simpler than the first approach involving transcoding. For example, when conducting a fourfold search, the throughput of the transcoding approach involves 1.552 Gbit/s for decoding, while the parsing throughput requires only 60 Mbit/s, which is almost a factor of 26 lower. This difference is so large that it is evident to select the second system concept.

The linear bandwidth increase with the growing trick-play playback speed can become significant for higher speeds. For example, Table 3.2 shows that for a playback speed of 11, the bandwidth becomes 165 Mbit/s compared to 30–60 Mbit/s for low search speeds. This linear bandwidth increase can be avoided, to bound the throughput rate as much as possible. When utilizing the random-access capability of the storage medium, it is possible to selectively address the intra-coded pictures at a nearly individual basis. This random access is possible for both disc-based storage (BD) and solid-state memory (SSD) storage. In this approach, only normal-play fragments are retrieved containing intra-coded pictures, avoiding full bit-stream parsing at high data rates to extract these I-pictures. A solution to achieve this objective is the usage of Characteristic Point Information (CPI), a method which is also employed by the Blu-Ray (BD) optical disc standard [89]. CPI is a separately stored information signal, containing metadata descriptions of the stored audiovisual program. An example of such metadata is the start location of I-pictures on the storage medium. In this thesis, this type of information is indicated as *locator informa-*

tion, revealing the start position of a particular access unit for decoding.

The second concept with bounded bandwidth for fast search based on meta-data, forms the basis for a cost-efficient video navigation solution, suitable to be deployed on standard DTV platforms.

3.4 Networked full-frame video navigation

In this section, we propose two algorithms suitable for performing autonomous client-server-based video browsing over a short-time interval, addressing fast-search playback in forward or reverse direction and slow-motion playback in forward direction. The two algorithms are based on the second concept, employing full re-use of MPEG-2-compressed normal-play pictures in combination with an MPEG-2-compliant rate control, as discussed in Section 3.3(B). Hereby, the MPEG-2-compliant rate control is different from a traditional rate control, as it is based on repetition pictures, which simultaneously control the picture frame rate as well as the bit rate.

Prior to commencing with the fast-search and slow-motion video navigation algorithms, we introduce the elegant solution of repetition pictures, which are based on the MPEG predictive-coded pictures (P or B). Furthermore, we elaborate on essential system aspect related to the re-use of compressed normal-play pictures and the perceptual impact of repetition pictures on the final video navigation.

3.4.1 Usage of repetition pictures

A *repetition picture* is an artificial duplicate of a previously decoded picture derived by means of prediction. This duplicate is inserted as a predictive-encoded picture in the compressed stream, with the purpose of manipulating the picture refresh-rate and/or bit rate of the video navigation signal, or creating an adaptation of the navigation playback speed. Manipulation of the picture refresh-rate and/or bit rate only occurs when generating the fast-search video navigation at the server-side of the network. Adaptation of the playback speed occurs for either fast-search or slow-motion playback. The adaptation of the playback speed and the two forms of rate control are on the basis of re-used compressed normal-play video sequences. The creation of repetition pictures for implementing fast search and slow motion is different for interlaced and progressive video formats, while it also depends on the video navigation direction (forward/reverse), as depicted in Fig. 3.4. In television video, the video format is based on the following rules.

In *progressive* video, each video picture is a frame without distinction of odd and even lines and all lines are sampled at the same time instance. The

MPEG-2 coding is therefore frame-based. In *interlaced* video, a frame consists of two fields, each captured at a different time instance separated by halve of the frame period. A field contains either the odd or the even lines of the frame, which explains the terms “top” and “bottom” field.²

Let us now elaborate on the impact of field- and frame-based repetition pictures and the difference between uni- and bi-directionally-coded repetition pictures.

A. Interlace kill: a field-based repetition picture

In MPEG-2 video compression, temporal correlation is exploited using either field- or frame-based prediction. This feature can be elegantly employed for designing a video navigation system, with a smooth usage of predictive pictures based on field- and frame-based processing. More specifically, we will design a video navigation algorithm with repetition pictures, which enables rendering control at the field-by-field level, achieving a finer granularity of rendering control. This field-by-field rendering control is enabled for the situation that the normal-play video sequence has an interlaced video format. Furthermore, when the video scene contains significant motion, straightforward repetition of an interlaced picture may result in motion judder. This is particularly annoying for the viewer and should be avoided.

Special processing step for controlled field-elimination processing. A field-based repetition picture enables partial repetition of the reference buffer content by selecting a predefined reference field, eliminating the motion judder. This field-based elimination process is indicated as “*interlace kill*”. Figure 3.6 visualizes the impact of “*interlace kill*”. The picture in Fig. 3.6(a) is constructed from two fields, see Fig. 3.5. For example, the clearly moving football player is displaced during the field period, as shown in Fig. 3.6(b). When constructing a new picture on the basis of a single field, this displacement is removed, resulting in a static picture, see Fig. 3.6(c). This artifact removal comes at the expense of halving the vertical resolution. In this way, the field-elimination process is conducted by the MPEG-2 decoder.

B. Generation of repetition pictures using P- and B-picture syntax

For the repetition of re-used MPEG-2-compressed pictures, a distinction is made between normal-play reference pictures (I,P) and non-reference pictures (B).

²In analog broadcasting, sometimes the terms “odd” and “even” field are employed, referring to the polarity of the line numbers constructing that field, e.g. an odd field contains the odd lines etc.

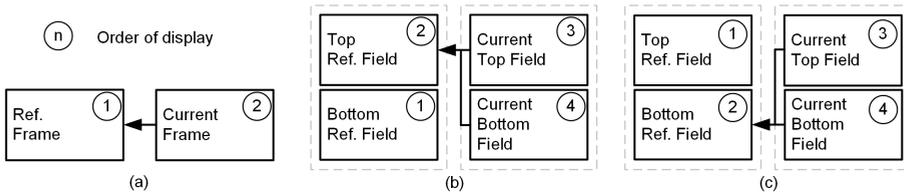


Figure 3.4 — Examples of creating predictive-coded repetition pictures. (a) Frame-based repetition picture for progressive video. (b) Repetition picture with “Interlace kill” by removing bottom-field for reverse direction trick-play. (c) Repetition picture with “Interlace kill” by removing top-field for forward direction trick-play.



Figure 3.5 — Interlaced picture consisting of (a) Top field and (b) Bottom field.

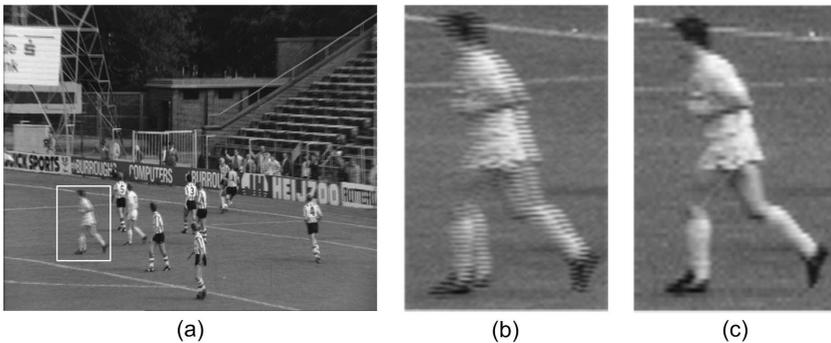


Figure 3.6 — Interlaced display. (a) Merged top and bottom field. (b) Football player constructed from interleaved top and bottom field, (c) Football player constructed with aid of two fields with equal content.

Repetition of reference pictures can be obtained via either P- or B-pictures. Although the visual effect is equal, the MPEG-2 decoder behavior is different for P- and B-pictures. When repetition is obtained using P-pictures, the reference buffer is updated with the reconstructed picture after decoding. This update of the reference buffer is not enforced when repetition is obtained utilizing B-pictures. These aspects result in a set of guidance rules for the final video navigation algorithm. A distinction is made between fast-search and slow-motion playback.

- *Normal-play progressive video / fast search and slow motion.* Due to the absence of motion in individual frames, each frame can be independently used for video navigation. Hence, both P- or B-pictures can be deployed for repetition of normal-play reference pictures. Repetition of normal-play B-pictures is obtained via re-decoding of that B-picture.
- *Interlaced video / fast search.* For fast search we use only I-pictures because of the high search speed. Repetition of I-pictures is always possible due to the absence of temporal dependencies. However, motion judder should be avoided by means of the “*Interlace kill*” processing step. Possible prediction pictures for repetition are P- and B-pictures.
- *Interlaced video / slow motion.* In slow motion, all pictures have to be decoded in order to preserve smooth motion. Hence, as a result all normal-play pictures have to be repeated, so that I-, P- and B-pictures will be used. However, due to the presence of motion in interlaced reference pictures, repetition of I-, P- and B-pictures will potentially result in visible motion judder. Unfortunately, the I- and P-reference pictures cannot be modified, since they serve as a reference for intermediate B- and future P-reference pictures. In order to avoid this judder, repetition of reference pictures in the decoder can only be achieved within the syntax rules of the MPEG-2 standard, which prescribe B-pictures for repetition at the current frame sequence position. For these pictures, the occurrence of motion judder can be avoided by employing “*interlace kill*”. Next to motion judder in reference pictures, motion judder can also be present in original B-pictures. Due to the coding nature of these pictures, extrapolation from past or future, or interpolation past and future, motion judder can only be avoided on the basis of transcoding, thereby eliminating one field. As this is computationally expensive, we propose to decode these pictures only once, thereby avoiding at least visible motion judder. Due to this rule, the smooth motion portrayal is jeopardized and a video navigation playback speed-error is introduced (normal-play B-pictures are not repeated).

3.4.2 System aspects for video navigation algorithms

Section 3.3 has proposed a cost-efficient video navigation concept, suitable to be deployed in a networked environment. This concept re-uses normal-play MPEG-2-compressed pictures, employing predictive-coded repetition pictures as discussed in Section 3.4.1.

This section elaborates on video system aspects caused by our proposal to re-use normal-play MPEG-2-compressed pictures for constructing a fast-search and slow-motion video navigation signal. One of these video system aspects involves the rate control of the MPEG-2 video system. Due to the insertion of repetition pictures, the influence on MPEG-2-compliant rate control involves both bit rate and video frame rate.

A. MPEG-2 rate-control adaptation when employing repetition pictures

Depending on the video format, MPEG-2 video compression may employ a different picture encoding method, see Fig. 3.7, which influences the derivation of a video navigation signal. Based on the applied MPEG-2 video coding, frame-based or field-based pictures are selected from a normal-play GOP to construct a fast-search video navigation signal. We have utilized this diagram for the derivation of a fast-search video navigation algorithm. The effect during fast-search playback is that inter-field motion is observed. It would be nice for fast-search that the moving object travels in the opposite motion direction, when performing backward search. However, this smooth motion portrayal in fast-search cannot be guaranteed beforehand. Although there is no guarantee, we can still positively influence the motion rendering during fast search by adjusting the top/bottom field rendering order, depending on the original motion direction. In practical MPEG-2 implementations, the bit costs of the individual I-, P- and B-pictures vary significantly. As a rule of thumb, an I-picture

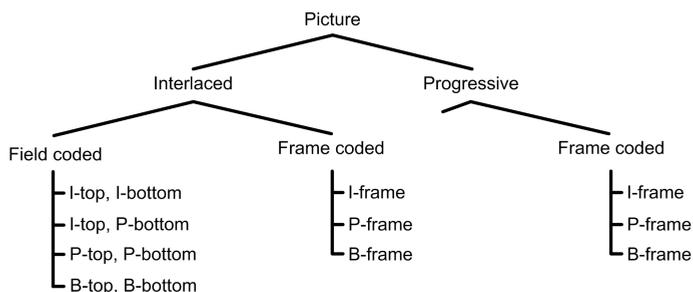


Figure 3.7 — Picture video format and MPEG-2 encoding options.

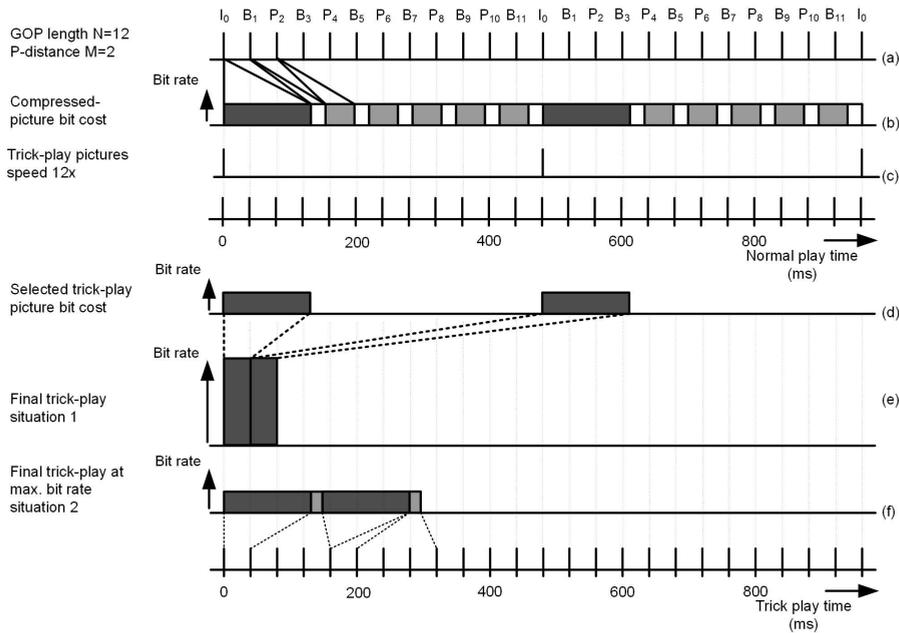


Figure 3.8 — *Low-cost MPEG-2-compliant fast-search trick play for 25-Hz television system. (a) Normal-play-compressed pictures. (b) Visualization of the used bit cost per compressed picture type and associated transmission time. Dark gray refers to an I-picture, white is a B-picture and medium gray is a P-picture bit-cost. (c) Temporal subsampling of the normal-play sequence, using a speed-up factor of 12, resulting in the selection of only I-pictures. (d) Retrieval of I-pictures at normal-play bit rate. (e) Trick-play bit rate due to re-used I-pictures at 25-Hz television rate. (f) Transmission of fast-search trick play based on re-used I-pictures and repetition pictures, indicated by medium gray color, to illustrate the ability to comply with the supported MPEG-2 Main Level for broadcasting.*

consumes the highest bit cost, while the P-pictures consume roughly one third of an I-picture and B-pictures require about one sixth up to one ninth of an I-picture.

Let us now return to the issue of MPEG-2 bit-rate control in the navigation framework of using repetition pictures. We have already indicated that both frame rate and bit rate (see above rule of thumb) are influencing our navigation concept. In the following discussion we will address both aspects by means of an example for deriving a fast-search video navigation signal.

Consider a typical MPEG-2-compressed video sequence as depicted in Fig. 3.8. This figure visualizes the impact of re-using normal-play I-pictures regarding the involved transmission time and corresponding bit rate. In Fig. 3.8(a), a typical MPEG-2 Group-Of-Pictures (GOP) structure is depicted, with an associated bit cost per picture indicated by Fig. 3.8(b). The transmission time of a re-used I-picture (see Fig. 3.8(c)) typically involves more than a single frame display period, see Fig. 3.8(d). When concatenating successive I-pictures selected from a normal-play sequence for fast-search playback, while employing the original normal-play transmission time, the resulting frame rate will be clearly lower. Such a decoder situation will probably lead to a decoder buffer underflow. Increasing the bit rate, such that the I-picture is transmitted in a single frame display period (see Fig. 3.8(e)), may exceed the maximum bit rate defined by the involved MPEG-2 *Level*, resulting in a bit-rate violation. In order to avoid a bit-rate violation and buffer underflow, we propose that repetition pictures [90] are inserted at the server side and transmitted (see Fig. 3.8(f)), prior to the (large) I-picture, thereby generating additional display time and thus transmission time. Although the usage of repetition pictures enables the derivation of an MPEG-2-compliant fast-search video navigation sequence, the effective navigation playback speed is reduced, resulting in a navigation playback speed-error. We will discuss this playback speed-error in the sequel.

B. Navigation playback speed-error

Video navigation on the basis of re-used MPEG-compressed normal-play video may suffer from a playback speed-error.

- **Fast-search playback speed-error.** A playback speed-error occurs during fast-search video navigation when using repetition pictures, to derive an MPEG-2-compliant fast-search video navigation sequence on either progressive or interlaced video.
- **Slow-motion playback speed-error.** A playback speed-error occurs during slow-motion on MPEG-2 Main-Profile-encoded³ interlaced video due to the absence of artifact-free picture repetition, when re-using MPEG-2-compressed normal-play B-pictures.

Fast-search playback speed-error. A fast-search playback speed-error arises when a selected I-picture requires a transmission time of more than one display period. Such an I-picture is preceded by repetition pictures resulting in a repetition of the last decoded I-picture. These repetition pictures lower the playback speed, resulting in a playback speed-error. Such a playback speed-error is avoided when skipping a number of normal-play I-pictures equal to the

³From now on, we assume that the compressed video is MPEG-2 MP encoded.

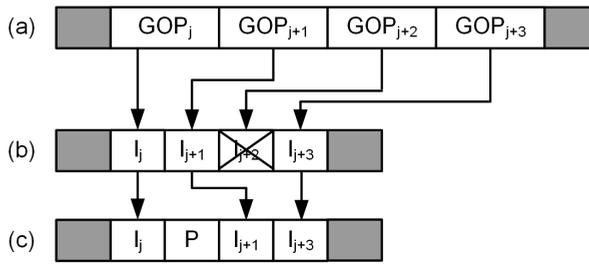


Figure 3.9 — Generation of fast-search trick play with reduced refresh-rate. (a) GOPs constructing the normal-play video. (b) Selected and skipped intra-coded pictures constructing a fast-search trick-play sequence. (c) Construction of a fast-search trick-play with P-picture based repetition resulting in an MPEG-2-compliant trick-play sequence.

number of inserted repetition pictures, succeeding the “large” I-picture. For the situation that I-pictures are not skipped, the insertion of repetition pictures allows the generation of a video navigation sequence with playback speed lower than the GOP length N .

Let us now further analyze an MPEG-2-encoded normal-play sequence with a GOP length of $N = 12$. Fast-search trick play is possible for playback speeds $P_s = \{12, 24, \dots, 12n\}$ with n being an integer, as discussed in Section 2.5.1. For the situation that a selected I-picture requires a transmission time of two display periods, a single repetition picture precedes such an I-picture, as depicted in Fig. 3.9. In Fig. 3.9, I-picture I_{j+1} requires a transmission time of two display periods. Repetition picture P , causes repetition of I-picture I_j , resulting in a time interval of two display periods to transmit I-picture I_{j+1} . The insertion of repetition picture P results in a playback speed-error, as effectively two images are derived from GOP_j instead of one, resulting in an effective playback speed $P_s = 6$ for the normal-play time interval corresponding to GOP_j . The intended playback speed is restored by skipping I-picture I_{j+2} , see Fig. 3.9.

Slow-motion playback speed-error. A slow-motion playback speed-error arises for the situation that an interlaced normal-play video sequence is encoded with Main-Profile MPEG-2. Such a video sequence contains B-pictures, which can only be displayed once, in order to avoid motion judder.

Let us now further analyze the repetition process in terms of the MPEG-2 GOP and MPEG-2 Profile parameters. For the situation of interlaced video, normal-play MPEG-2-compressed B-pictures are basically transmitted only once and not repeated, to avoid motion judder. The prevention of the above-mentioned motion judder, leads to a video navigation speed-error during slow motion,

caused by the fact that not every picture is displayed an equal amount of display periods. This situation can be avoided by displaying the reference pictures more often than required. This will become clear from the following analysis.

Let N be the GOP length and M the P-distance of an MPEG-2-compressed normal-play video sequence. The reciprocal of the trick-play slow-motion speed P_s is limited to an integer value, i.e. $1/P_s = \{2, 3, 4, \dots\}$. The number of display periods D_p , used for normal-play picture repetition during slow motion is specified by

$$D_p = \frac{1}{P_s}. \quad (3.1)$$

The product of $N \cdot D_p$ indicates the total amount of slow-motion pictures derived from a normal-play GOP. This set of slow-motion pictures can be separated into a sub-set based on repeated normal-play reference pictures, and a sub-set based on repeated normal-play B-pictures, which can be analytically described by

$$N \cdot D_p = D_p \cdot Ref_{pict} + D_p \cdot B_{pict}. \quad (3.2)$$

We define the *display error* D_e as the amount of absent display periods, which can be calculated on the basis of the normal-play GOP length N , P-distance M and the amount of display periods D_p used to repeat each normal-play picture. The *display error* is the difference between the intended display repetitions and the actual display repetitions and described by

$$D_e = N \cdot D_p - D_p \cdot \frac{N}{M} - (N - \frac{N}{M}). \quad (3.3)$$

Factorization of Eqn. 3.3 provides

$$D_e = N \left(D_p \left(1 - \frac{1}{M} \right) - \left(1 - \frac{1}{M} \right) \right). \quad (3.4)$$

From Eqn. (3.4), it becomes clear that for normal-play video sequences, encoded at MPEG-2 Simple Profile (SP), which is characterized by having $M = 1$, this results in $D_e = 0$, a zero-valued *display error*. For MPEG-2 MP with $M > 1$, the *display error* $D_e > 0$. The speed error can be compensated via additional display repetitions of the normal-play reference pictures. The number of reference pictures per GOP is N/M . The additional number of reference repetitions A_r is therefore the *display error* from Eqn. (3.4) divided by this fraction, resulting in

$$A_r = D_p(M - 1) - (M - 1). \quad (3.5)$$

For a normal-play video sequence with GOP lengths $N = 12$ and $N = 15$ and various values of M , Table 3.3, shows the *display error* and additional reference repetitions for a slow-motion speed of $1/3$. For the situation that the reciprocal

of the applied slow-motion playback speed (P_s) is an integer value, the slow-motion playback speed-error compensation on the basis of A_r , see Eqn. (3.5), the additional repetition of reference pictures, results in an exact speed-error compensation.

We now proceed with an example on the derivation of a slow-motion sequence on the basis of re-used MPEG-2-compressed pictures. Figure 3.10(a) indicates a fragment from an MPEG-2-compressed normal-play sequence. Figure 3.10(b) shows a derived slow-motion navigation sequence for a playback speed of $1/2$. This slow-motion video navigation sequence is obtained by means of reference picture repetition on the basis of artificially derived repetition pictures and re-decoding of B-Pictures. Such a derivation of a slow-motion video navigation sequence is possible for a video sequence with a progressive format. Figure 3.10(c) indicates the slow-motion fragment derived from the normal-play fragment depicted in Fig. 3.10(a), for the situation that the normal-play video sequence has an interlaced format. The video navigation sequence depicted in Fig. 3.10(c) has a speed error, as not all normal-play pictures are equally repeated. Figure 3.10(d) depicts the slow-motion video navigation sequence, which has been compensated for a trick-play speed-error, on the basis of additional reference picture repetitions.

Collected from Section 3.4.1 and Section 3.4.2, we summarize the main design rules for MPEG-2-compliant fast-search and slow-motion video navigation.

- The video navigation sequence follows a push-based communication model.
- Repetition pictures are deployed for trick play, while realizing MPEG-2-

Table 3.3 — Display error D_e and additional reference repetitions A_r for $D_p = 3$ and various values of M with $N = 12$ and $N = 15$.

N	M	N/M	D_e	A_r
12	2	6	12	2
12	3	4	16	4
12	4	3	18	6
12	6	2	20	10
15	2	6	15	2
15	3	4	20	4
15	4	3	22.5	6
15	6	2	25	10

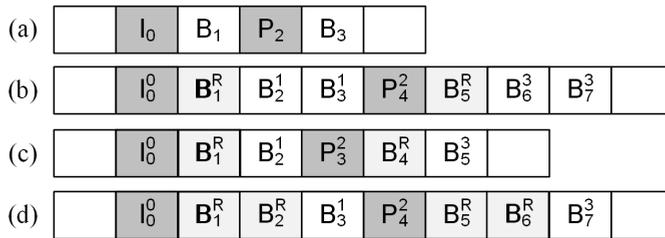


Figure 3.10 — Examples of slow-motion behavior for interlaced and progressive MPEG-2-compressed video with halved playback speed ($1/2$). (a) Normal-play video GOP with $N = 4, M = 2$. (b) Derived slow motion for the situation of progressive video. (c) Derived slow motion for the situation of interlaced video with playback speed-error. (d) Derived slow motion for the situation of interlaced video with correct playback speed.

compliant rate control.

- Both progressive and interlaced video formats are supported.
- “Interlace kill” for field-based rendering control is adopted, to avoid motion judder.
- The rendering order of fields is adapted to forward or reverse search direction.
- Trick-play playback speed-errors are compensated.

3.4.3 Algorithm for MPEG-2-compliant fast-search trick play

This section presents our proposed algorithm for deriving an MPEG-2-compliant fast-search trick-play sequence, for either forward or reverse search on the basis of re-used MPEG-2-compressed normal-play I-pictures. The algorithm is based on the design rules discussed in Section 3.4.1 and Section 3.4.2.

Figure 3.11 depicts the video navigation framework facilitating fast-search and slow-motion. The involved signal processing for both forms of video navigation is separated in two steps.

The first step is conducted during recording, while the second step is performed during video navigation playback. In the first step, the recorded program is analyzed, deriving essential bitstream aspects, called Characteristic Point Information (CPI), which are stored as metadata on the storage medium. In the second step, signal processing is conducted on the provided normal-play information, resulting in an MPEG-2-compliant fast-search or a slow-motion

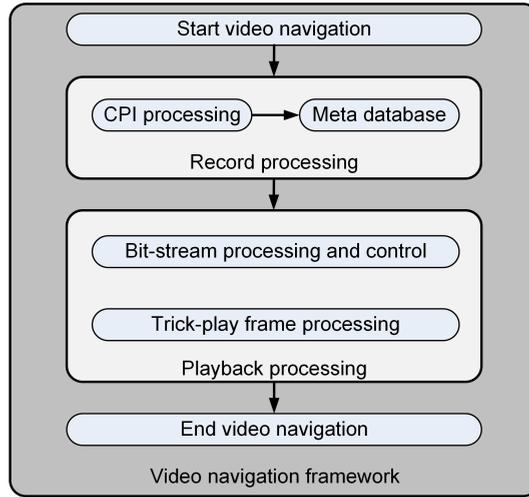


Figure 3.11 — *Fast-search and slow-motion video navigation framework.*

video navigation signal. During fast-search, fragments of the full normal-play TS containing the I-picture are provided to the playback processing, while during slow-motion, the full normal-play TS is provided at a playback speed equal to the slow-motion speed P_s . Next to the visual information, CPI information is supplied to the playback processing, simplifying the generation of the final MPEG-2-compliant video navigation signal. We will now elaborate on the algorithm of our proposed video navigation.

CPI generation during record

The proposed fast-search and slow-motion video navigation algorithms employ characteristics of the MPEG-2-compressed normal-play video sequence, which cannot all be retrieved from the MPEG-2 syntax. In order to avoid stream analysis during video navigation playback, the normal-play video sequence is analyzed during recording. For each normal-play GOP, the following characteristics are calculated: (1) the GOP length N , (2) the P-distance M , (3) number of repetition pictures to transmit the I-picture and (4) the I-picture start position in the recorded TS. Appendix D shows the algorithmic steps for deriving these CPI components.

Assumption on interlaced video broadcast

In Section 3.4.1, the concept of “interlace kill” has been introduced, where “odd” (top) and “even” (bottom) fields were distinguished. In the sequel we assume that interlaced video always starts with the “odd” (top). The proposed algo-

rithms are based on this format, which sometimes will be modified depending on the search direction in order to optimize the perceptual quality of the navigation. When modified, this will be explicitly indicated in the algorithms. In practice, this is a reasonable assumption, as broadcasters typically maintain this format.

Fast-search signal processing during playback

Figure 3.12 indicates the flowchart revealing the fast-search playback processing to generate a fast-search video navigation sequence. The video navigation processing involves readout of the MPEG-2 headers preceding the I-picture (Parse MPEG-2 headers), revealing the normal-play video format (progressive or interlaced) and picture size (width and height). On the basis of the derived picture size, a repetition picture is generated. The generation of a repetition pictures is detailed in Fig. 3.13 and should be seen as an insert (Generate repetition picture) to Fig. 3.12. In case of an interlaced video format (`interlaced == True ?`), access to the picture header is required in order to adapt the rendering order of top and bottom field, depending on the video navigation direction (`reverse == True ?`). Insertion of repetition pictures (`nr_rep_pict > 0 ?`) depends on the I-picture metadata, which contains the number of repetition pictures “`nr_rep_pict`” (Read `nr_rep_pict`). This number was already calculated and stored as metadata during recording and reveals the number of required repetition pictures to be generated and transmitted prior to an I-picture, creating an MPEG-2-compliant video navigation sequence. Adaptation of the “`nr_rep_pict`” value may be required (Calculate `speed-error`) when deriving a fast-search video navigation sequence for playback speeds $P_s < N$, with N being the normal-play GOP length and P_s the trick-play playback speed. This adaptation is incorporated in the detailed algorithmic description, see Algorithm 1 and Algorithm 2. Figure 3.13 shows the flowchart for generating a repetition picture, employed in the flowchart depicted in Fig. 3.12. In this figure, two signal paths are depicted. At the left, the repetition picture generation involved for progressive video is depicted. The generation depends only on the picture size (width, height). The second signal path indicates the repetition picture generation for interlaced video format. The difference is based on the reference field from which the prediction is made, which depends on the video search direction (`reverse == True ?`). Slices forming a repetition picture are constructed on the basis of the normal-play image width, while the normal-play image height determines the amount of slices constructing a repetition picture. For the situation that a video signal originates from an interlaced video source, the repetition picture is cre-

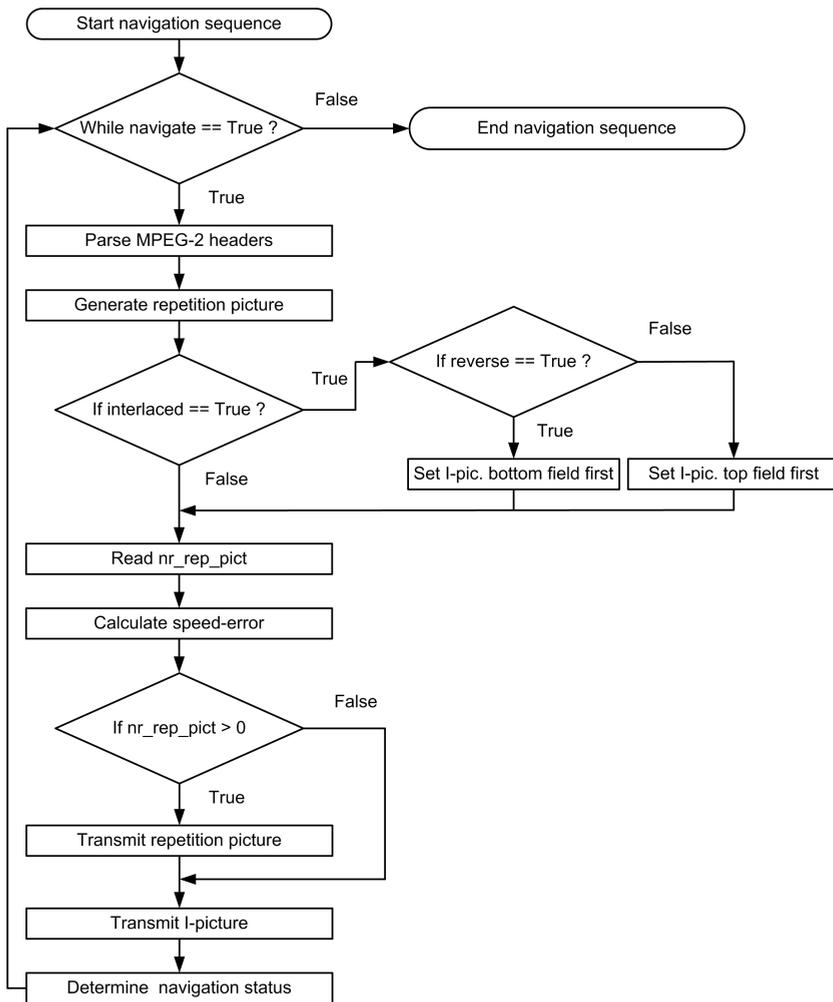


Figure 3.12 — *Fast-search video sequence generation re-using MPEG-2-coded I-pictures for fast-search video navigation in forward or reverse direction.*

ated on the basis of two field-based pictures, both referring to the same reference field (top/bottom). In this way, motion judder is avoided (“interlace kill”), whereby the eliminated field depends on the search direction, as depicted in Fig. 3.4(b)(c). Details on the fast-search video navigation algorithmic steps are found in Algorithms 1, 2 and 3. These algorithms are not related to Fig. 3.13.

Algorithm 1 indicates the algorithmic steps involved in selecting I-pictures

Algorithm 1 I-pictures selection for fast-search trick play

Require: P_s , $metadata[]$, $direction$ **Ensure:** I-picture selection and correct nr_rep_pict value for fast-search trick play at playback speed P_s **Initialize:** $CPIindex = 0$, $NextCPIindex = 0$, $Speed_error = 0$ **while** not end of fast search **do** \triangleright Retrieve selected I-picture and CPI $N = metadata[CPIindex].N$ \triangleright read GOP length $nr_rep_pict = metadata[CPIindex].nr_rep_pict$ \triangleright read nr_rep_pict **if** ($P_s < N$) **then** $Add_rep = \lfloor N/P_s \rfloor - 1$ \triangleright calculate rep. pict. per GOP**if** ($CPIindex \bmod P_s == 0$) **then** \triangleright det. speed correction point $Speed_error = (N \bmod P_s)$ \triangleright calculate speed error**end if****else** $Error = (P_s \bmod N)$ \triangleright calculate speed error $SumError = +Error$ \triangleright cumulative speed error**if** ($\lfloor SumError/P_s \rfloor == 1$) **then** $SumError = -P_s$ **if** $direction == forward$ **then** $NextCPIindex = NextCPIindex + 1$ \triangleright compensate speed

error

else $NextCPIindex = NextCPIindex - 1$ **end if****end if****end if****if** ($CPIindex == NextCPIindex$) **then****switch** P_s **do****case** ($P_s < N$)call $SelectIpic_1()$ **case** ($P_s \geq N$)call $SelectIpic_2()$ **end if****if** $direction == forward$ **then** $CPIindex = +1$ \triangleright go to next CPI entry**else** $CPIindex = -1$ \triangleright go to previous CPI entry**end if****end while**

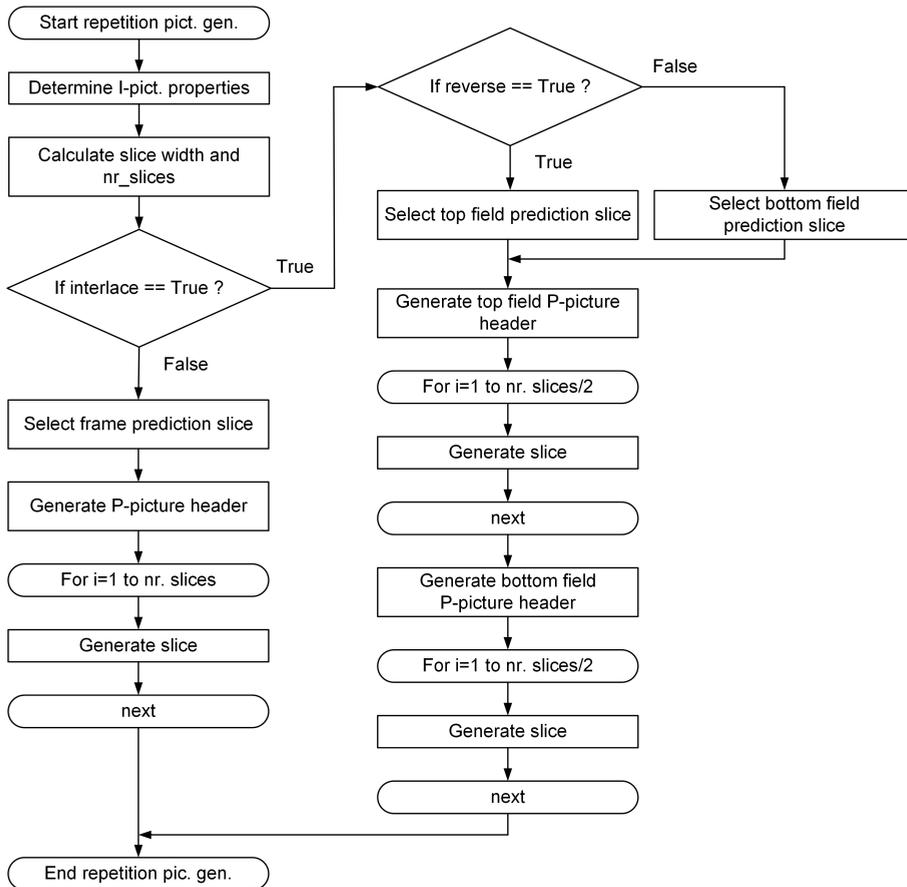


Figure 3.13 — *Repetition picture generation for fast-search forward or reverse video navigation.*

constructing the video navigation sequence, as depicted in Fig. 3.12, with the objective to avoid a video navigation playback speed-error. Such an error may occur, when the re-used MPEG-2-compressed I-pictures are located at different temporal locations, as imposed by the fast-search video navigation temporal subsampling grid. Correction of this playback speed-error is distributed over the Algorithms 1, 2 and 3. Hereby, Algorithm 1 performs playback-speed tracking over time (fine-tune operation), while Algorithm 2 and Algorithm 3 implement a coarse navigation speed adaptation. However, the playback speed-error correction conducted by Algorithm 3 may be exact, all depending on the navigation playback speed P_s and normal-play GOP length N .

Algorithm 2 Select I-picture for $P_s < N$

```

function SELECTIPIC.1()
  if ( $nr\_rep\_pict > Add\_rep$ ) then
     $temp1 = nr\_rep\_pict - Add\_rep$            ▷ calc. rep. pict. difference
     $temp2 = temp1 \text{ div } (Add\_rep + 1)$ 
     $temp3 = temp1 \text{ mod } (Add\_rep + 1)$ 
     $temp4 = (Add\_rep + 1) - temp3$          ▷ calc. remaining rep. pict.
    if ( $temp3 == 0$ ) then
       $skip\_CPI = temp2$                    ▷ skip nr. GOP
    else
       $skip\_CPI = temp2 + 1$                ▷ skip nr. GOP
       $nr\_rep\_pict = +temp4$                ▷ adjust navigation GOP
    end if
    if ( $direction == forward$ ) then
       $NextCPIindex = CPIindex + skip\_CPI$    ▷ skip GOP entries
    else
       $NextCPIindex = CPIindex - skip\_CPI$    ▷ skip GOP entries
    end if
  else
    if ( $direction == forward$ ) then
       $NextCPIindex = CPIindex + 1$          ▷ next GOP entry
    else
       $NextCPIindex = CPIindex - 1$          ▷ previous GOP entry
    end if
     $nr\_rep\_pict = Add\_rep$                ▷ compensate for absent images
  end if
  retrieve TS fragment                       ▷ read I-picture
   $nr\_rep\_pict = +Speed\_error$            ▷ compensate for speed error
   $Speed\_error = 0$                        ▷ reset speed error
end function

```

Algorithm 1 calculates a potential playback speed-error, which depends on the relation between the normal-play GOP length N and the fast-search playback speed P_s . This calculation is performed on the CPI data (stored as meta-data), during navigation playback. Depending on the outcome of the relation between the navigation playback speed P_s and the normal-play GOP length N , either Algorithm 2 or Algorithm 3 is applied.

There are two approaches for compensating the playback speed-error. For the situation that $P_s < N$, a playback speed-error occurs when $N \bmod P_s \neq 0$, while for $P_s \geq N$, a playback speed-error occurs when $P_s \bmod N \neq 0$. A playback speed-error can be reset to zero at discrete moments in time or averaged

over time. To illustrate this, Algorithm 1 employs both methods. For the situation that $P_s < N$, the playback speed-error is set to zero at discrete moments in time, while for the $P_s \geq N$ situation, the speed-error is averaged over time. The motivation for this approach is that for the latter situation, a playback speed-error reset involves skipping multiple successive I-pictures, which results in a loss of navigation information. For the situation $P_s < N$, the speed-error compensation results in a longer display time of pictorial information.

Algorithm 2 is applied when $P_s < N$. For this fast-search playback situation, the algorithm solves the main inequality $nr_rep_pict > Add_rep$. Hereby, nr_rep_pict indicates the amount of repetition pictures required to transmit an I-picture, while Add_rep indicates the amount of repetition pictures required to reduce a navigation playback speed-error. Solving inequality $nr_rep_pict > Add_rep$ results in calculation of the next CPI entry, indicated by $NextCPIindex$, as well as the final value for nr_rep_pict . This inequality is solved by employing a concept of skipping a successive I-picture ($skip_CPI$), which is enabled by generating a navigation GOP length N , being a multiple of the value $Add_rep + 1$. Here, the value $Add_rep + 1$ reveals the minimum navigation GOP length, that matches the video navigation playback speed. This calculation is conducted employing four intermediate variables $temp1, \dots, temp4$, see Algorithm 2. The variable $temp1$ indicates the difference in repetition pictures that exist between nr_rep_pict and Add_rep . The calculated difference $temp1$ is tested against the navigation GOP length $Add_rep + 1$. The value of $temp2$ is used to calculate the next CPI entry, thereby skipping an I-picture, whereas the value of $temp3$ is used to control the adaptation of the final amount of repetition pictures. For this adaptation, an error signal $temp4$ is calculated, indicating the additional involved repetition pictures, creating a navigation GOP with a GOP length that is an integer multiple of $Add_rep + 1$.

Algorithm 3 is applied when $P_s \geq N$. The objective is to reduce a potential speed-error. This algorithm calculates, depending on the value of nr_rep_pict , the next CPI entry. The amount of GOP's that are skipped is indicated by $P_s \text{ div } N$. This integer division value may be scaled by nr_rep_pict , for the situation that $nr_rep_pict > 0$. The calculation, $P_s \text{ div } N$, is exact for the situation $P_s \text{ mod } N = 0$. When the calculation $P_s \text{ mod } N \neq 0$, a playback speed-error remains, which is solved in Algorithm 1, as discussed earlier.

3.4.4 Algorithm for MPEG-2-compliant slow-motion trick play

This section presents our proposed algorithm to derive an MPEG-2-compliant slow-motion video navigation sequence for forward search, re-using MPEG-2-compressed normal-play pictures. The algorithm considers system aspects discussed in Section 3.4.2 and employs repetition pictures, as discussed in Section 3.4.1, to realize the required playback frame-rate control. Figure 3.11 de-

Algorithm 3 Select I-picture for $P_s \geq N$

```
function SELECTIPIC.2()
  if (nr_rep_pict > 0) then
    if (direction == forward) then
      NextCPIindex = CPIindex + (nr_rep_pict ·  $\lfloor P_s/N \rfloor$ )
    else
      NextCPIindex = CPIindex - (nr_rep_pict ·  $\lfloor P_s/N \rfloor$ )
    end if
  else if
    if (direction == forward) then
      NextCPIindex = CPIindex +  $\lfloor P_s/N \rfloor$            ▷ Next entry
    else
      NextCPIindex = CPIindex -  $\lfloor P_s/N \rfloor$            ▷ previous entry
    end if
  end if
  retrieve TS fragment           ▷ read I-picture
end function
```

picts the video navigation framework already discussed, but now employed for slow-motion video navigation. The involved slow-motion signal processing is separated into two steps, as described in Section 3.4.3, resulting in signal processing during recording and signal processing during slow-motion playback. During recording, the P-distance M is explicitly calculated in advance, thereby avoiding bitstream analysis at the picture header level during slow-motion playback. The proposed algorithm is suitable to derive a slow-motion video navigation sequence for both progressive and interlaced video format, either coded in MPEG-2 Simple or Main Profile or MPEG-2 intraframe. Slow-motion video navigation is obtained by repetitive display of the individual pictures constructing a normal-play video sequence. When conducting the repetition process on the basis of re-used MPEG-2-compressed pictures, the relation between successive compressed pictures has to be considered. Figure 3.14 visualizes the possible transitions between the three different MPEG-2 picture types, constructing a normal-play video sequence. Figure 3.15 depicts the slow-motion signal processing flowchart, revealing the main processing steps, which are different for interlaced and progressive scan video. The employed repetition pictures are generated (Generate repetition picture) prior to slow-motion video navigation. Fig. 3.16 should be seen as an insert to the previous Fig. 3.15 (Generate repetition picture) and describes either a P-repetition picture or a B-repetition picture. Our proposed algorithm employs the transitions between successive compressed pictures as visualized in Fig. 3.14, to control the derivation of the final slow-motion MPEG-2-compliant

video navigation sequence. Algorithm 4 shows algorithmic steps detailing the overall slow-motion flowchart in Fig. 3.15. Depending on the parsed picture (**switch curpict do**), either Algorithm 5, 6 or 7 is applied. Let us now further elaborate on Algorithm 4. During slow-motion playback, the normal-play video sequence is provided at a playback speed corresponding to the slow-motion playback speed P_s (now a fraction), resulting in a reduced frame rate $f_r \cdot P_s$, with f_r being the normal-play television frame rate. The involved slow-motion processing either inserts MPEG-2-compressed repetition pictures, or repeats re-used normal-play pictures, as shown in Algorithm 5, 6, 7, re-establishing the full television frame rate f_r . Insertion of repetition pictures or repetition of the current picture n , depends on the properties of the current picture n , the previous picture $n - 1$ and in case of interlaced format also on the P-distance M . Parameter n is an incremental index indicating MPEG-2-compressed pictures in transmission order. Prior to filtering the properties of picture n by means of a filter operation in the MPEG-2-compressed domain, the TS is first demultiplexed, see at the top of Algorithm 4. The obtained video ES is parsed to locate individual pictures, filter the requested parameters and adapt top/bottom field rendering. The extracted parameters are the video format and the image size, while the rendering processes adapted is controlled (*video format == interlaced*) such that a top field is always rendered first. The processing encompasses start code detection conform the MPEG-2 standard and retrieving the full compressed picture. On the basis of picture

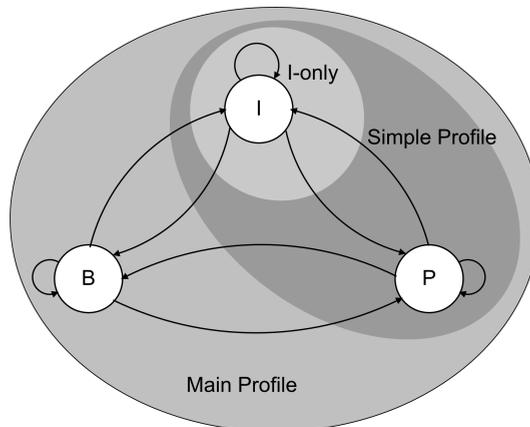


Figure 3.14 — Possible picture transitions of an MPEG-2-coded normal-play video sequence.

n (**switch curpict do**) and picture $n - 1$, the algorithm decides to insert repetition pictures, repeating the previous reference picture, or duplicating picture n (B-picture), see Algorithm 5, 6, 7. The slow-motion algorithm utilizes either P-pictures or B-pictures to repeat the previous reference picture, see Algorithm 8. For the situation that the normal-play video sequence is intraframe-coded, I-picture repetition is conducted on the basis of inserting P-pictures, whereas slow-motion for MPEG-2 SP and MP involves B-pictures for reference picture repetition. For progressive video, B-pictures in MPEG-2 MP are repeated by means of repetitive decoding, while for interlaced video, B-pictures are decoded only once, in order to avoid motion judder, see Algorithm 9. Motion

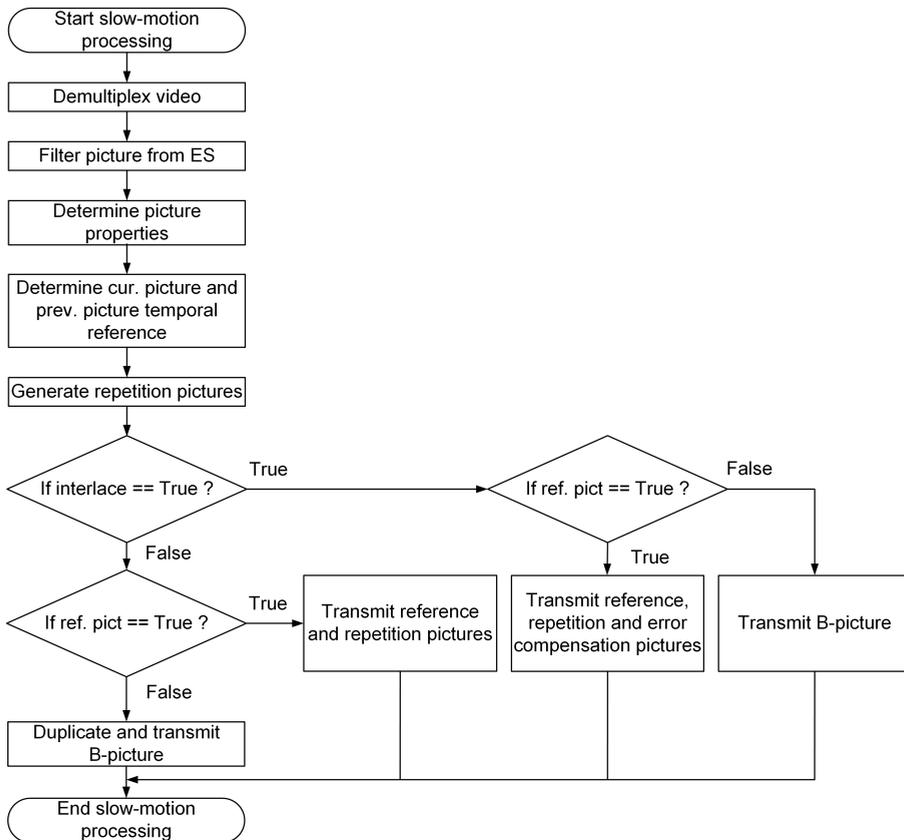


Figure 3.15 — Flowchart slow-motion playback signal processing re-using MPEG-2-coded normal-play pictures.

Algorithm 4 MPEG-2-compliant slow-motion trick play**Require:** $P_s, P_{distance}$ **Ensure:** Slow-motion for playback speed P_s **Initialize:** $prevpict = Null, D_p = 1/P_s$

```

while not end of slowmotion do
  demultiplex TS packets
  find start code in video ES
  if (start code == sequence header) then
    get picture size;
    transmit parsed data
  end if
  if (start code == sequence extension) then
    video format = progressive sequence
    transmit parsed data
  end if
  if (start code == GOP header) then
     $M = P_{distance}$ ;
    transmit parsed data
  end if
  if (start code == picture header) then
    curpict = picture type
    curtempref = temporal reference
    if (video format == interlaced) then
      top field = 1
    end if
    parse current picture
  end if
  switch curpict do
    case I
      call I-picture()
    case P
      call P-picture()
    case B
      call B-picture()
  end switch
   $prevpict = curpict$ 
   $preftempref = curtempref$ 
end while

```

judder is avoided by employing repetition pictures enforcing “Interlace kill”, therefore the top/bottom-field rendering order of interlaced pictures is set such

that always the top field is rendered first. In this way, the “Interlace kill”-based repetition pictures will always repeat the last displayed field. Prior to explaining the algorithmic details of slow-motion signal processing, we discuss the meaning of the temporal reference index number assigned to each compressed

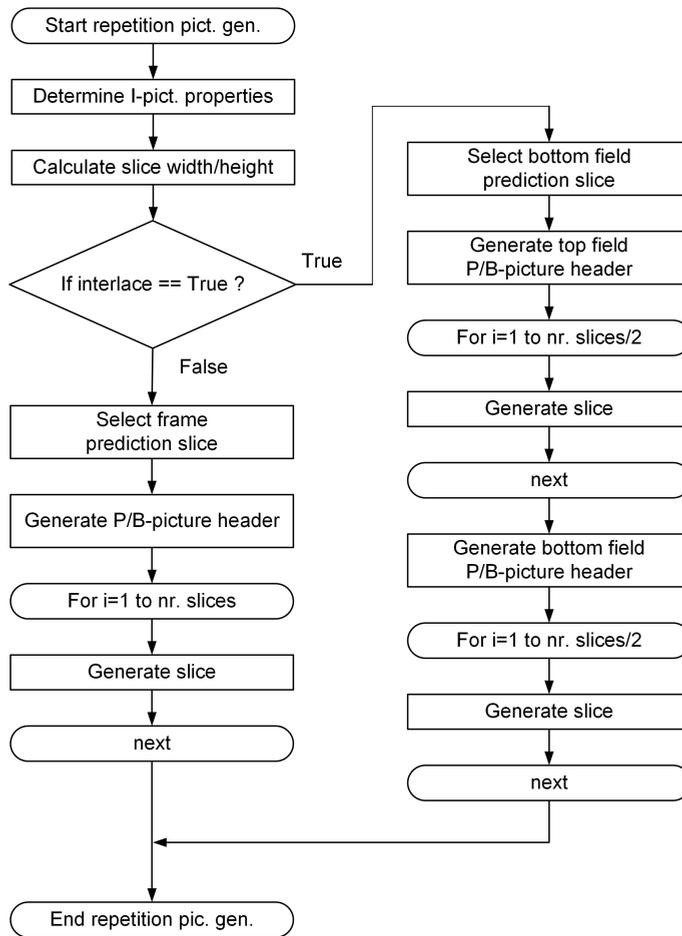


Figure 3.16 — Repetition picture generation for slow-motion forward video navigation based on P- or B-type picture syntax. For repetition, from normal-play intra-only MPEG-2 sequences, the I-pictures are repeated as P-pictures. For repetition, normal-play MPEG-2 sequences using SP or MP coding employ only B-pictures. Hence, the figure should be read as either P- or B-picture only, which depends on the video format.

Algorithm 5 I-picture slow motion processing

```

function I-PICTURE()
    temporal reference = 0
    transmit current picture
    start_navigation = False
    switch prevpict do
        case (prevpict == Null)
            start_navigation = True                                ▷ potential open GOP
        case (prevpict == I)
            temporal reference = prevtempref + 1                    ▷ I-only sequence
            call P-repetition()
            start_navigation = False
        case (prevpict == P)
            temporal reference =  $D_p * \textit{prevtempref} + 1$ 
            call B-repetition()
            start_navigation = False
        case (prevpict == B)
            temporal reference =  $D_p * (\textit{prevtempref} + 1) + 1$ 
            call B-repetition()
            start_navigation = False
    end function

```

MPEG-2 picture. Due to the absence of the P-distance M in the MPEG-2 syntax, the MPEG-2 video decoder determines M on the basis of the temporal index distance between two successive reference frames. Since we are creating additional pictures for slow-motion, our algorithm has to satisfy the MPEG-2 syntax and therefore should calculate the correct temporal index, combined with the newly employed repetition picture type. In this way, the correct calculation of the decoding and presentation time stamps is facilitated at the server and can be flawlessly used by the decoder client.

Algorithm 5 reveals the four algorithmic steps for the situation that the current picture is an I-picture. Slow-motion video navigation always starts with an I-picture, enabling proper sequence decoding. For a normal-play *open* GOP situation, which becomes clear when parsing the next normal-play picture, a signal *start_navigation* is set. This signal enables removal of potential B-picture(s), which may not be correctly decoded, due to the absence of a previous reference picture, see Algorithm 7 (**if** *start_navigation* == *True* **then**). The signal *start_navigation* is only active until the next reference picture is processed, see Algorithm 5 and Algorithm 6. For the situation that the previous picture equals *Null* (**case** *prevpict* == *Null*), this indicates the very beginning of a slow-motion

Algorithm 6 P-picture slow-motion processing

```
function P-PICTURE()
    temporal reference =  $D_p * curtempref$ 
    transmit current picture
    switch prevpict do
        case (prevpict == I)
            temporal reference = curtempref
            call B-repetition()                ▷ duplicates previous ref. pict.
        case (prevpict == P)
            temporal reference =  $D_p * prevtempref + 1$ 
            call B-repetition()                ▷ duplicates previous ref. pict.
        case (prevpict == B)
            temporal reference =  $D_p * (curtempref - M) + 1$ 
            call B-repetition()                ▷ duplicates previous ref. pict.
    start_navigation = False                ▷ no more B-picture need to be removed
end function
```

video navigation sequence. In this exceptional case, the current I-picture is used to start the navigation sequence. Technically, this means that the signal *start_navigation* is set "True", the temporal reference is set and the current I-picture is transmitted. For the regular situation that we are processing, we detect a transition between successive compressed picture types, in order to determine the correct slow-motion processing. When the previous picture type equals I (**case** *prevpict* == I), P-repetition pictures are employed as a construct, to make a repetition based on the previous I-picture (see Algorithm 8), followed by the transmission of the current I-picture. For the situation that the previous picture type equals P (**case** *prevpict* == P), while the current type is I, the temporal reference is set, the I-picture is transmitted followed by the transmission of locally derived B-based repetition pictures (B-type is the expected type after I-type), see Algorithm 8. These B-pictures effectively repeat the previous P-picture. For the situation that the previous picture type equals B (**case** *prevpict* == B), the temporal reference is set and the current I-picture is transmitted. In order to repeat the previous reference picture, which is presumably a P-type picture, we insert B-based repetition pictures as expected, which enforces repetition of this last P-type reference picture, see Algorithm 8.

Algorithm 6 reveals the three algorithmic steps for the situation that the current picture is a P-picture. When the previous picture type equals I (**case** *prevpict* == I), B-repetition pictures are employed to repeat the previous I-picture (see Algorithm 8). Prior to repetition picture insertion, the P-picture is transmitted with a new temporal reference. For the situation that the previous picture

Algorithm 7 B-picture slow-motion processing

```

function B-PICTURE()
  switch prevpict do
    case (prevpict == I)
      temporal reference =  $D_p * curtempref - 1$ 
      if start_navigation == True then
        remove B-Picture                                ▷ avoid decoding error
      else
        call B-duplication()                            ▷ duplicates current picture
      end if
    case (prevpict == P)
      temporal reference =  $D_p * curtempref$ 
      call B-duplication()                              ▷ duplicates current picture
    case (prevpict == B)
      if (video format == interlaced) then
        temporal reference = temporal reference      ▷ value already
        calculated
      else
        temporal reference =  $D_p * curtempref$ 
      end if
      if start_navigation == True then
        remove B-Picture                                ▷ avoid decoding error
      else
        call B-duplication()                            ▷ duplicates current picture
      end if
  end function

```

type equals P (**case** *prevpict* == *P*), while the current type is P, the temporal reference is set, the current P-picture is transmitted followed by the transmission of locally derived B-based repetition pictures, see Algorithm 8. These B-pictures effectively repeat the previous P-picture. For the situation that the previous picture type equals B (**case** *prevpict* == *B*), the temporal reference is set and the current P-picture is transmitted. In order to repeat the previous reference picture, which can either be an I-type or a P-type picture, we insert B-based repetition pictures, which enforces repetition of this last reference picture, see Algorithm 8.

Algorithm 7 reveals the three algorithmic steps for the situation that the current picture is a B-picture. When the previous picture type equals I (**case** *prevpict* == *I*), a test is conducted (**if** *start_navigation* == *True* **then**) to determine whether the current B-picture can be removed, otherwise the current B-picture is employed. For the situation that the video format is progressive, this B-type

Algorithm 8 P/B-repetition picture

```
function PB-REPETITION()
   $A_r = D_p(M - 1) - (M - 1)$  ▷ cal. additional rep. pict.
  if (video format == progressive) then
    for  $i = 1$  to  $D_p - 1$  do
      insert P/B-picture ▷ P- or B-type repetition picture
      temporal reference = ++
    end for
  else
    for  $i = 1$  to  $D_p - 1$  do ▷ for B-picture loop range is  $D_p - 1 + A_r$ 
      insert P/B-field top picture ▷ P- or B-type repetition field
      insert P/B-field bottom picture ▷ P- or B-type repetition field
      temporal reference = ++
    end for
  end if
end function
```

Algorithm 9 B-duplication picture

```
function B-DUPLICATION()
  transmit current picture
  temporal reference = ++
  if (video format == progressive) then
    for  $i = 1$  to  $D_p - 2$  do
      transmit current picture
      temporal reference = ++
    end for
  end if
end function
```

picture is repeated $D_p - 1$ times, whereas for the interlaced situation this picture is employed once, see Algorithm 9. This repetition processing applies to the next two cases. For the situation that the previous picture type equals P (**case** *prevpict* == *P*), while the current type is B, the temporal reference is set, the current B-picture is transmitted. For the situation that the previous picture type equals B (**case** *prevpict* == *B*), a test is conducted (**if** *start_navigation* == *True* **then**) to determine whether the current B-picture can be removed, otherwise the current B-picture is employed.

Algorithm 8 reveals the repetition picture insertion for both P- and B-pictu-res. For the situation that the normal-play video sequence is intraframe

encoded, P-pictures are employed for picture repetition, whereas for MPEG-2 SP and MP video coding, B-pictures are employed for repetition. The employed repetition pictures follow the employed video format (progressive/interlaced) that they are based upon, where we use “*Interlace kill*” for interlaced video. Furthermore, for the interlaced video format situation, in order to avoid a video navigation speed-error, the reference pictures are additionally repeated according to the value of A_r .

Algorithm 9 reveals the B-picture duplication. For normal-play progressive video, B-pictures are repeated conform the slow-motion playback speed, whereas for interlaced video, each normal-play B-picture is only rendered once, avoiding the appearance of motion judder.

3.5 Conceptual plane-based MPEG-2-compliant video navigation

In this section, we provide a conceptual outline for solving MPEG-2-compliant plane-based video navigation over a network, which can be realized in two ways. Both solutions follow the video navigation concept of a thumbnail-based mosaic screen⁴, whereby the thumbnails are derived on the basis of temporal resampling of the normal-play video sequence. The first solution conducts all involved signal processing during video navigation, while the second solution is based on re-use of MPEG-2-compressed thumbnail-sized video information, derived during recording. For both solutions, a conceptual architecture is presented including performance estimation figures for key system resources, revealing the complexity of both solutions.

A. First concept: Spatial-domain construction of mosaic screen

In the first approach, a video navigation solution employing a mosaic screen is constructed on the basis of fast-search video information, derived from an MPEG-2-compressed normal-play video sequence. In such a framework, we use MPEG-2 decoding, scaling, MPEG-2 encoding and MPEG-2 packetizing, see Fig. 3.17(a). In this figure, at the left side, MPEG-compressed data is retrieved from the storage medium and passes through an MPEG-2 demux to access the video ES, containing the individually encoded I-pictures. In order to derive thumbnail-sized pictures, the video ES is fully decoded, resulting in decompressed I-pictures. These pictures are downsampled to thumbnail-sized images and employed to construct the final mosaic-screen, which is MPEG-2

⁴In this thesis, we have adopted the term mosaic screen as proposed earlier in literature by the author. However, due to the desired minimum size of the employed thumbnail, the term mosaic screen may be better replaced by the term tile-based screen.

Table 3.4 — *Performance estimation on system resource utilization for mosaic-screen construction based on 16 thumbnail-sized images derived from SD video.*

Concept	System resources					
	Decoder Mb/s	Scaler Mb/s	Mosaic Screen Mb/s	Encoder Mb/s	Memory Capacity kBytes	DSP / CPU cycle load
Spatial- domain construction	89	85	-	388	2,656	High
Compressed- domain construction	11	11	1.7	0.8	1,109	Low

Simple Profile (SP) encoded. Finally, this video navigation sequence is packetized forming a new MPEG-compliant stream, which can be transmitted across a network to any type of client. In the previously described process, the mosaic-screen is constructed in the spatial domain, which explains the term spatial-domain construction.

Considering the involved signal processing associated with the concept for deriving a mosaic-screen video navigation signal, enables us to derive a performance estimation and thus the computational load on the key system resources. The outcome of this estimation is summarized in Table 3.4. Video navigation on the basis of mosaic screens is a manually controlled navigation form, for which we deploy an update rate of 1 Hz, to generate a new mosaic screen. The figures in Table 3.4 apply to generating a mosaic screen that is based on 16 thumbnail-sized images at an update rate of 1 Hz, derived from a 720×576 picture size (SD) and 4:2:0 sampling format for the video signal and I-only decoding.

Using previous figures, an image with 16 thumbnails is constructed from 16 SD images with the above resolutions and sampling format. This results in an uncompressed video bit rate of 79.63 Mbit/s and involves a memory capacity requirement of 608 kBytes per full-color frame ($720 \times 576 \times 1.5$ pixels). The bit rate for such an SD signal and the resulting mosaic screen at 25 Hz (full frame rate) equals 124.42 Mbit/s.

The load on the system resources is further increased due to the involved MPEG-2 demultiplexing and multiplexing of the compressed video ES into an MPEG-2-compliant TS.

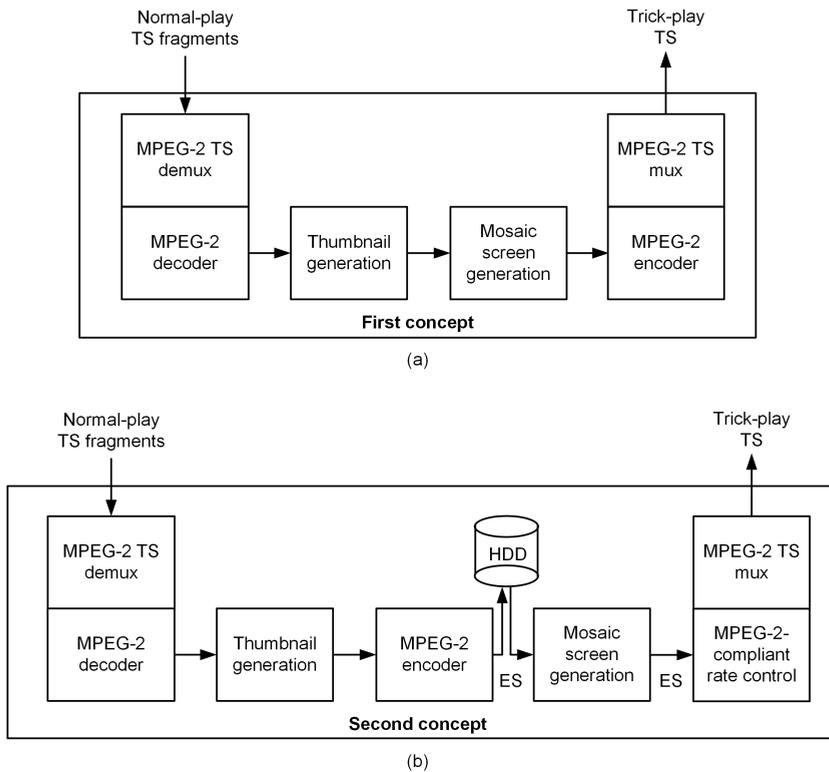


Figure 3.17 — Conceptual signal processing flow for video navigation based on mosaic screens. (a) Construction of mosaic screen in spatial domain. (b) Signal processing concept for construction of mosaic screens in the MPEG-2-compressed domain using MPEG-2-compressed thumbnail-sized pictures.

B. Second concept: Compressed-domain construction of mosaic screen

In a second approach, a mosaic-screen navigation sequence is constructed in the MPEG-2-compressed domain on the basis of MPEG-2 subpictures. These subpictures are derived from an MPEG-2 normal-play video sequence invoking: MPEG-2 decoding, scaling, MPEG-2 encoding, subpicture storage, mosaic-screen composition and finally, the generation of a rate-controlled video navigation sequence, see Fig. 3.17(b). In this figure, at the left side, MPEG-2-compressed data is received from a digital broadcast and passes through an MPEG-2 demux to access the video ES, containing the individually encoded I-pictures. In an MPEG-2 video broadcast, the distance between successive I-pictures equals the GOP-length N , which is typically 12 pictures. For European broadcast-

ing, this results in two I-pictures per second, which significantly reduces the involved throughput rate. In order to derive subpictures of the desired size, the video ES is fully decoded, leading to decompressed I-pictures. These pictures are then downsampled, MPEG-2 encoded and stored on the Hard Disk Drive (HDD). The mosaic screen is now constructed in the MPEG-2-compressed domain, while the final video navigation sequence is rate-controlled on the basis of repetition pictures. This navigation sequence is packetized forming a new MPEG-2-compliant TS, which can be transmitted across a network to any type of client.

Taking into account the involved signal processing associated with the concept for deriving a mosaic-screen video navigation, allows us to estimate the computational load and throughput rate based on the key system resources. The outcome of this estimation is summarized in the same table as the spatial-domain solution, see Table 3.4.

Let us apply a similar reasoning as in the first case to quantify some system requirements. Again, we apply an update rate of 1 Hz, for generating a new mosaic screen, while employing 16 MPEG-2-compressed subpictures per mosaic screen. The subpictures are derived from a 2-Hz SD video sequence with a resolution of 720×576 pixels and 4:2:0 color sampling format. This yields in an uncompressed video bit rate of 9.95 Mbit/s and involves a memory-capacity requirement of 608 kBytes per full-color frame. For a derived subpicture size of 180×144 pixels and 4:2:0 sampling format, the uncompressed subpicture bit rate equals 0.62 Mbit/s and involves a memory capacity of 38 kBytes per full-color frame. After MPEG-2 compression with a factor three, the 16 compressed subpictures constructing a single mosaic screen result in a bit rate of 1.65 Mbit/s.

Similar as in the spatial-domain situation, the load on system resources is further increased when including the involved MPEG-2 demultiplexing and multiplexing of the compressed video ES into an MPEG-2-compliant TS.

When comparing the two proposed concepts with respect to their system requirements, it becomes clear that the second navigation concept based on mosaic-screen with compressed subpictures results in a significantly lower throughput rate than the first concept. Moreover, the second navigation concept has a more balanced bandwidth consumption, so that a high bandwidth peak is avoided. For these reasons, we adopt the second system approach.

3.6 Networked hierarchical mosaic-screen navigation

In this section, we propose an algorithm suitable for performing manual client-based video browsing over a long-time interval on the basis of hierarchical mosaic screens.

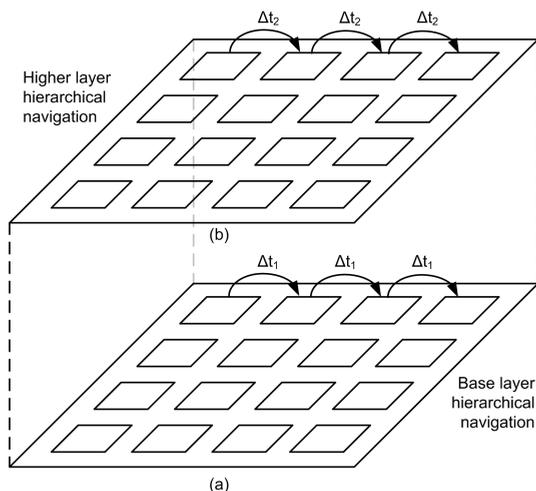


Figure 3.18 — Layered mosaic screens constructing hierarchical navigation layers.

Prior to commencing with the hierarchical navigation algorithm, we introduce the concept of hierarchical mosaic screens, followed by essential system aspects involved with the derivation of subpictures and the final generation of the mosaic screen on the basis of re-used MPEG-2-compressed subpictures.

3.6.1 Concept of hierarchical mosaic-screen navigation

In the previous section we have elaborated on the derivation of subpictures constructing a mosaic screen. This section concentrates on employing such screens in a hierarchical manner, in order to provide efficient video navigation, which offers an instantaneous video summarization at different temporal scales. Figure 3.18 shows the concept of hierarchical video navigation on the basis of layered mosaic screens. The video navigation summarization strength is determined by the subpictures constructing the individual hierarchical mosaic screens, more particularly the differences between the employed subpictures. Figure 3.18 shows the concept of hierarchical mosaic screens. The instantaneous overview of the base layer depends on the value Δt_1 , while for the higher layer this depends on Δt_2 . An actual value for both Δt_1 and Δt_2 can either be constant or variable. When the value for $\Delta t_2 \gg \Delta t_1$, the summarization time interval of a higher navigation layer is significantly increased. The difference between successive subpictures of a mosaic screen may be based on the temporal content changes between successive scenes in a video program. This is due to the fact that a program director constructs a program from a set

of different scenes. One of the construction rules is that the director uses a certain amount of scene changes per time interval to create e.g. an atmosphere of liveliness, continuous action and story telling. This means that every scene has typically an average duration. When knowing this average scene duration, a subpicture from each scene can be obtained by equidistant temporal subsampling. The mosaic screen is then constructed from these sampled subpictures.

A good starting point for further discussion is the temporal correlation of a regular video sequence. The temporal correlation heavily depends on the nature of the video material, resulting in typical scene durations of 3–5 seconds. This observation forms the basis for the selection of the normal-play I-pictures, from which the subpictures are derived to construct the base-layer mosaic screens. Furthermore, on the basis of this typical scene duration, we can derive the number of hierarchical navigation layers. When using an average of 3 seconds for scene duration, a mosaic screen consisting of 16 subpictures presents an instantaneous video information overview of 48 seconds of normal-play video information. Navigating through hierarchical mosaic screens corresponds to a temporal zoom-in zoom-out operation. For this type of navigation, it is attractive to employ three navigation layers offering an instantaneous overview varying from 1 minute for the base layer, up to a quarter of an hour for the middle navigation layer and up to a few hours for the third navigation layer. Evidently, this approach requires a subpicture selection algorithm revealing an appropriate summary of the video program and its constituting scenes. Mosaic-screen construction based on re-used MPEG-2 subpictures involves a two-step approach. The first step involves the selection of subpictures composing the mosaic screen, while the second step contains signal processing yielding an MPEG-2-compliant video navigation sequence.

The subpicture selection part of the algorithm will be discussed in Section 3.6.3. In the next section we elaborate on system aspects related to subpicture derivation, mosaic screens construction based on re-use of MPEG-2 subpictures and viewer feedback information supporting the navigation with mosaic screens.

3.6.2 System aspects of mosaic-screen navigation algorithm

Section 3.5 has proposed a cost-efficient plane-based video navigation concept, applicable to a networked environment. The concept re-uses subpictures derived from normal-play MPEG-compressed I-pictures in combination with predictive-coded repetition pictures, as discussed in Section 3.4.1, to provide the required rate control resulting in an MPEG-compliant navigation sequence. Our desire to re-use MPEG-compressed subpictures to construct a mosaic screen, imposes specific signal processing and conditions that need to be aligned with the MPEG-2 syntax and/or compliance rules. The concept of a mosaic screen constructed on the fly from re-used MPEG-2 pictures, see Fig. 3.19, en-

forces constraints regarding the subpicture coding. This implies the following aspects.

- *Scalable MPEG-2 intraframe video decoding.* From the normal-play video sequence, subpictures are derived using the intraframe-compressed pictures. As subpictures have a lower resolution than normal-play pictures, scalable decoding in the DCT-domain is beneficial, and is therefore pursued, thereby avoiding an additional video scaling operation at the pixel level after decoding.
- *MPEG-2 coding of subpictures.* The MPEG-2 subpictures are based on a coding method using so-called “mini-slices”. This enables re-use of the MPEG-compressed video information for constructing a mosaic screen without transcoding. Furthermore, similarity between successive subpictures allows to perform predictive coding on those subpictures. This enables on-the-fly construction of mosaic screens.
- *Mosaic-screen construction in the MPEG-2-compressed domain.* When constructing mosaic screens in the compressed domain on the basis of subpictures, a number of measures have to be taken to generate an MPEG-compliant navigation sequence in the form of a mosaic screen. These measures involve a.o. mini-slice adaptation, MPEG-header generation and rate control.
- *On Screen Display.* On screen display in a networked environment requires transcoding in order to insert video overlay information, such as a progress indicator for navigation. Navigation within one mosaic screen or between several screens requires a visual starting point from which further navigation is conducted. This starting point is indicated by means of a bounding box around the considered subpicture.

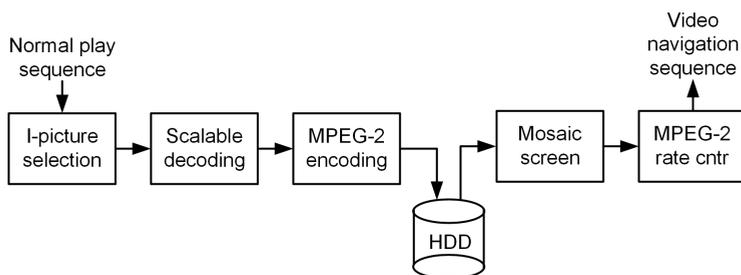


Figure 3.19 — Mosaic screen signal processing chain.

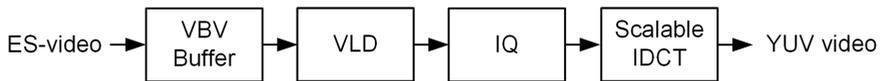


Figure 3.20 — *Main building blocks of the scalable MPEG-2 intraframe decoder.*

Let us now further elaborate on each of the above-mentioned aspects.

A. Scalable MPEG-2 intraframe video decoding

The plane-based video navigation concepts discussed in Section 3.5 involves an intraframe decoding operation, followed by a scaling operation to derive the final subpictures and their appropriate size. As these two operations occur in consecutive order, it is beneficial to combine them into a single computational step, known as (complexity) scalable decoding [88], [91]. This type of scalable decoding is based on selecting available, already-coded coefficient information to perform the decoding solely based on the selected information. This reduces the amount of computations at the decoder, thereby limiting the utilization of system resources. This approach avoids the calculation of intermediate signals at a resolution larger than the target resolution, so that the resulting throughput rate is reduced. For example, let us consider the construction of a mosaic screen in the compressed domain using 16 subpictures and SD MPEG-2 video decoding. Scalable decoding would require a throughput bandwidth of 1.8 Mb/s, instead of 11 Mb/s in the non-scalable way. Moreover, the required bandwidth for video scaling would become zero. Consequently, the throughput rate for MPEG-2 decoding is reduced by approximately a factor 6, compared to the values depicted in Table 3.4, without any further scaling bandwidth.

Let us now detail the algorithm of a scalable MPEG-2 decoder. Since the derived subpictures originate from intraframe-compressed pictures, the scalable decoder is based on the conventional intraframe decoding steps, depicted in Fig. 3.20. Spatial subsampling in the MPEG-compressed domain is achieved by modification of the 2D-IDCT. Figure 3.21 indicates the maximal received MPEG-2 DCT coefficients of which a selection is made for applying the scalable MPEG-2 IDCT. The scalable 2D-IDCT uses only 4 coefficients, resulting in a 2×2 pixel block. The transform of 2×2 coefficients converts a Standard-Definition (SD) picture into a Quarter Common Intermediate Format (QCIF) picture (176×144 pixels). Equation (3.6) presents the regular MPEG-2 two-

dimensional IDCT which is specified by

$$f(x, y) = \frac{2}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} C(u)C(v) \cos\left(\frac{(2x+1)\pi u}{2N}\right) \cos\left(\frac{(2y+1)\pi v}{2N}\right), \quad (3.6)$$

$$C(u), C(v) = \begin{cases} \frac{1}{\sqrt{2}} & \text{for } u, v=0, \\ 1 & \text{otherwise.} \end{cases}$$

for $x, y, u, v = 0, 1, 2, \dots, N-1$, where x, y are the pixel coordinates in the picture domain and u, v are coordinates in the transform domain. A standard MPEG-2 decoder employs 8×8 blocks ($N = 8$), while for the scalable MPEG-2 SD-to-QCIF decoder this value becomes $N = 2$. For a scalable MPEG-2 SD-to-QCIF decoder, the computation of Equation (3.6) is visualized in a diagram in Figure 3.22 for $N = 2$. Figure 3.22 indicates two different implementations for computing the scalable 2D IDCT. The first implementation (a) is a direct translation of Equation (3.6), whereas the second implementation (b) is a simplified version, in which multiplications have been removed and replaced by shift operations. In this case, the normalization of the coefficients is modified to $1/8$, so that shifting is enabled. For intraframe decoding this introduces a small pixel error, which is not noticeable.

The result of this scalable decoding operation as discussed above is a subpicture extracted from the intraframe picture (see Fig 3.19) that is decoded in a scalable fashion, leading to the intended subpicture format.

Let us now derive the computational speed-up of our proposed scalable 2D IDCT. The two 2×2 IDCT implementations depicted in Fig. 3.22 were benchmarked on an ARM926 processor running at 160 MHz. The influence of bus

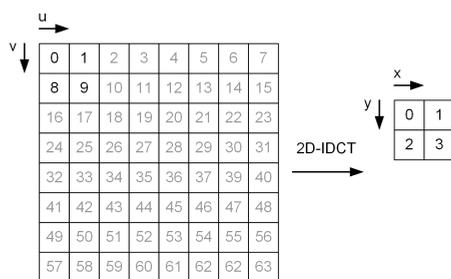


Figure 3.21 — Scalable 2×2 IDCT for an SD-to-QCIF MPEG-2 decoder. At the left side, the received coefficients (0..63) are depicted. At the right side, a 2×2 pixel block is depicted obtained by applying a 2×2 scalable IDCT.

latency is eliminated, by continuously executing the test routine on the same data set. The test routine containing the IDCT is invoked 9,720 times, which is equivalent to the maximum number of DCT blocks of an MPEG-2 ML intraframe picture with a 4:2:0 color sampling format. For a scalable decoder using multiplications, the execution time is 14.8 ms and consumes 2.32 Mcycles. The simplified IDCT using shift operations requires 8.5 ms and 1.36 Mcycles. This simplified algorithm almost halves the computational complexity.

We conclude that for a European television system employing a 40 ms frame period, the proposed simplified 2×2 IDCT consumes only 3.4 % of the total frame period.

B. MPEG-2 coding of subpictures

In order to encode a subpicture in MPEG-2, the subpicture is divided into slices, which are composed of a consecutive set of macroblocks, as discussed in Section 2.2.1. The division of a subpicture into slices and macroblocks has to be considered, when constructing a mosaic screen in the MPEG-2-compressed domain. The subpicture size is chosen such that it corresponds to an integer number of macroblocks both horizontally and vertically. According to MPEG-2 coding, a horizontal row of macroblocks forms a slice. Limiting the length of a slice to the width of a subpicture creates a so-called “mini-slice” with respect to the mosaic screen. For example, with four subpictures in the horizontal direction of a mosaic screen, we create four mini slices horizontally in the mosaic screen (see Figure 3.23). The MPEG-2 syntax to code a mini slice depends on the employed picture coding type for encoding a mosaic screen. MPEG-2 video coding allows either intraframe coding, uni-directional predictive coding or a combination of these two coding methods.

Let us now discuss the MPEG-2 coding aspects of our proposal. The ob-

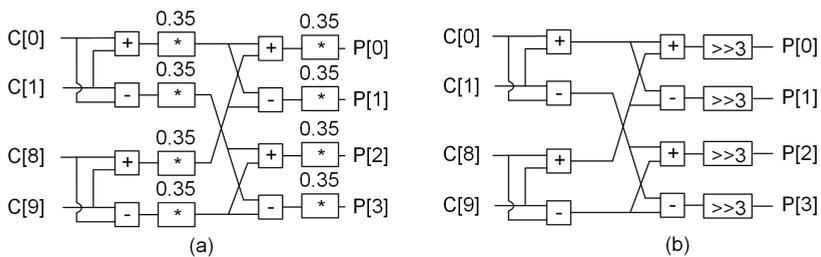


Figure 3.22 — Scalable 2×2 IDCT for an SD-to-QCIF MPEG-2 decoder. (a) Standard IDCT implementation. (b) Optimized IDCT implementation removing the coefficient multiplications.



Figure 3.23 — *Construction of a mosaic screen using subpictures composed of mini-slices.*

jective is to encode the subpictures facilitating the mosaic-screen application while avoiding an expensive implementation. This is achieved by selecting a proper picture coding type in combination with an elegant mosaic-screen composition. The usage of a predictive picture type for compression allows motion-compensation techniques at a low computation cost, thereby reducing the computation cycles at the applied processor. Since each subpicture is a frame-based selection from an existing video sequence, each subpicture is independently coded, resulting in intraframe-coded macroblocks and thus intra-coded mini-slices. We have employed MPEG-2 compression for the mini-slices, leading to MPEG-2 subpictures. Although each mini-slice is composed of intraframe-encoded macroblocks, an MPEG-2 subpicture, and thus a mosaic screen, can be of various picture types. MPEG-2 offers three picture types (I, P and B). A straightforward choice would be to select the I-type picture. However, this requires that each mosaic screen is fully re-composed and decoded for each screen, even if a significant similarity exists between two time-succeeding mosaic screens. Since this processing is typically implemented in software, aiming at execution on a small co-processor, we avoid expensive computations at this processor, and thus re-use of already available information at the decoder side.

As a result, we propose to use the syntax of a P-type picture for encoding the mini-slices, thereby exploiting forward motion compensation. Furthermore, we propose to encode each mini-slice in a fixed bit cost within the MPEG-2 framework. The usage of predictive coding enables to re-use the already decoded information from a previous screen to construct the actual mosaic screen. The correct filling of a motion-compensated macroblock is obtained by defin-

Subpicture 1	Subpicture 2	Subpicture 3	Subpicture 4
Subpicture 5	Subpicture 6	Subpicture 7	Subpicture 8
Subpicture 9	Subpicture 10	Subpicture 11	Subpicture 12
Subpicture 13	Subpicture 14	Subpicture 15	Subpicture 16

Figure 3.24 — *Spatial location ordering of subpictures in a mosaic screen on the basis of intra- or inter-coded subpictures.*

ing a motion vector that refers to the desired information. Hence, the MPEG-2 decoder performs motion compensation for those areas that already exist with a zero difference signal (skip macroblock), and adds missing subpictures via intraframe decoding.

The fact that all elements of a mosaic screen are MPEG-2 compliant, e.g. mini-slices and subpictures, allows for MPEG-2 re-encoding of the constructed mosaic screen. This would enable browsing through neighboring mosaic screens which are on-the-fly constructed. The use of P-coded mini-slices enables smooth scrolling operations that can easily be facilitated by a computing core in conjunction with the available MPEG-2 decoding. This novel extension will be elaborated at the end of this chapter.

C. Mosaic-screen construction in the MPEG-2-compressed domain

Mosaic-screen construction based on re-used MPEG-2-compressed subpictures in accordance with the method described in Section 3.6.1 requires adaptation of the full-screen MPEG-2-compressed data in order to aid the decoder in understanding the format of a mosaic screen consisting of subpictures constructed from mini-slices. We summarize the most important adaptations in the sequel.

- *Automatic spatial positioning of subpictures.* The mini-slices constructing the subpictures have to be correctly positioned and aligned to form the final subpicture-based mosaic screen. Therefore, we modify the slice headers of the mini-slices, in particular the *slice_start_code* which indicates the new vertical position in the mosaic screen. Similarly, the horizontal position is adapted by modification of the *macroblock_address_increment*. Both parameters are part of the MPEG-2 video standard.

- *Initialization of an MPEG video decoder.* An MPEG video decoder requires initialization to properly decode a “mini-slice” based video sequence. This initialization information is contained in a set of headers, preceding the coded mini-slices constructing a mosaic screen. In our case for the construction of a mosaic screen, we employ two headers: the sequence and picture header and their corresponding extensions. This means that we construct an MPEG-compliant video sequence containing the mosaic-screen.
- *MPEG-compliant rate control.* In order to generate a decodable video sequence, i.e. an MPEG-compressed video navigation sequence, repetition pictures are employed as discussed in Section 3.4.1. The resulting bit stream should be MPEG-compliant requiring also rate control. We adopt the same algorithm from fast-forward navigation for the creation of our mosaic screen navigation sequence. There are three differences: (1) the mosaic screen is always made from progressive video, so that extensions for interlacing can be skipped, (2) the mosaic screen is coded in P-pictures instead of I-pictures, (3) the number of repetition pictures required to create sufficient transmission time for communicating a mosaic screen is not generated during recording, but is selected as a fixed setting depending on the desired quality of the mosaic screen.

D. On Screen Display

Navigation through a set of mosaic screens involves optical feedback to the user. Typically, such an optical feedback is in the form of a bounding box surrounding a subpicture, indicating the actual navigation position. The visualization of this bounding box is called On Screen Display (OSD). In a network-based environment, there are three approaches to conduct OSD. In dual-channel video, the OSD information is transmitted as a second video information signal, which is mixed at the client-site. This requires a dual-channel video decoder, which is not standardized. In a second approach, the MPEG-2-compressed subpicture is transcoded, involving full decoding, video mixing at pixel level and re-encoding of the mixed video signal. This is too expensive in terms of computations.

The third and preferred option, which is also adopted, is to generate each subpicture in two formats: one with a bounding box and one without a bounding box. Hence, all subpictures are recorded in both formats as metadata. During navigation, the appropriate selection of subpictures is made. This solution doubles the involved storage requirements, but avoids the computational load during navigation. However, the encoding load during recording is not doubled, by elegantly sharing the compressed information of the inner part of the compressed subpicture between the two different picture formats.

3.6.3 Algorithm for MPEG-2-compliant mosaic-screen navigation

This section presents our proposed algorithm for generating a navigation sequence based on mosaic screens, re-using MPEG-2-compressed subpictures. The generation of a navigation sequence based on mosaic screens follows a two-step approach, involving signal processing during (1) recording and (2) video navigation. The algorithm is based on the chosen system aspects discussed in Section 3.6.2 and employs repetition pictures, as discussed in Section 3.4.1, to realize the final video navigation sequence.

A. Subpicture selection algorithm for hierarchical mosaic screen navigation

In this section, we propose the subpicture selection algorithm for constructing the hierarchical mosaic screens. We propose the usage of three hierarchical levels to summarize normal-play fragments, employed at different time scales.

Pictures constructing a digital video sequence can be denoted by $f_n(i, j)$, for $0 \leq j \leq W - 1$ and $0 \leq i \leq H - 1$, where n is the frame index, running from 0, 1, 2... till the end of the normal-play sequence. The parameter W represents the picture width and H denotes the picture height. The subpictures of the base layer are derived from the sequence $f_n(i, j)$ by applying a subsampling of frame index n , where the subsampling factor corresponds to the average scene duration. This factor is explained below and differs for various video navigation layers.

Let us assume that n corresponds to the time nT_f , where T_f is the frame time, which e.g. equals 40 ms (1/25 sec.) for the European case. Our approach is to apply a selective subsampling within the sequence of frames. The subsample factor for doing this is defined by parameter k_s , which equals the speed-up factor for fast-search video navigation. Using the observation that a video scene on the average changes every three seconds and one subpicture of each scene is desired, the base-layer subsampling factor k_s amounts to $k_s = 3 \times 25 = 75$. Equation (3.7) indicates the frame index k_1 that selects the frames from the normal-play sequence $f_n(i, j)$. These selected images form the base-layer for the mosaic screen. The selected frame indexes are specified by

$$k_1 = nk_s \text{ for } k_s = 75, n = 0, 1, 2, \dots, n_{\max}. \quad (3.7)$$

This parameter selection results in 1,200 subpictures per hour of normal-play video, leading to 75 mosaic screens containing 16 subpictures at the base layer, each summarizing 48 seconds of normal-play video. A typical two-hour movie is summarized in this way by 150 mosaic screens. With these numbers, interpretation of the individual mosaic screens becomes a tedious task, especially when hours of video material are involved. This situation is avoided when applying hierarchical layering of mosaic screens. The hierarchical structure allows

efficient high-speed browsing through a complete movie and enables a quick identification of the sequence of interest. Our hierarchical navigation proposal employs three layers. The relation between the hierarchical layers is chosen such that, when descending in the hierarchy, there is a well-defined relation between the subpictures of the higher layer and the mosaic screens of the adjacent lower layer. Therefore, any higher layer mosaic screen is derived from the base-layer subpictures, using a subsampling factor that is an integer multiple of the base-layer subsampling factor. In this way, only one subpicture of the lower layer mosaic screen appears in the next adjacent higher layer mosaic screen. Equation (3.8) indicates the frame index for the subpictures selection process of the second hierarchical navigation layer, which is specified by

$$k_2 = nk_s S \text{ for } k_s = 75, S = 16, n = 0, 1, 2, \dots n_{\max}. \quad (3.8)$$

In this expression, S is the number of subpictures per mosaic screen and k_2 the frame index of the second hierarchical navigation layer. Likewise, Equation (3.9) indicates the frame index for the subpictures selection process of the third hierarchical navigation layer, resulting in

$$k_3 = nk_s S^2 \text{ for } k_s = 75, S = 16, n = 0, 1, 2, \dots n_{\max}. \quad (3.9)$$

Here k_3 is the subsampling factor for the third hierarchical navigation layer. The final subpicture selection for layer l becomes now such that from the original picture sequence $f_n(i, j)$, the following frames $f_{kl}(i, j)$ for $l = 1, 2, 3$, are selected:

$$\begin{cases} l = 1 : \text{ base layer } & f_{k1}(i, j), \\ l = 2 : \text{ second layer } & f_{k2}(i, j), \\ l = 3 : \text{ third layer } & f_{k3}(i, j), \end{cases} \quad (3.10)$$

where the picture indexes $k1, k2, k3$ are specified as in Equations (3.7), (3.8) and (3.9).

The above picture subsampling grid has omitted one system aspect that we consider now for refinement. The subpictures constructing a mosaic screen are derived from normal-play *I-type* compressed pictures, which occur at a repetition rate of the GOP size N . Consequently, in order to properly select the I-pictures, we redefine the base-layer subsampling factor k_s to be an integer multiple of the GOP length N , while incorporating the average scene duration. With an average scene duration of 3 seconds and a typical GOP length $N = 12$, the value for $k_s = 72$. Figure 3.25 visualizes the normal-play summary per mosaic screen for each hierarchical layer and $k_s = 72$.

B. Signal processing during recording

In order to support mosaic-screen video navigation, specific extra signal processing during recording is conducted, as proposed in Section 3.5. As this vi-

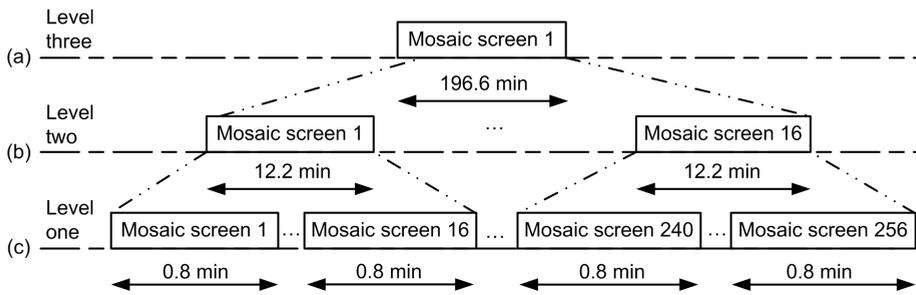


Figure 3.25 — Hierarchical navigation layers, their normal-play abstract duration and absolute normal-play picture index for mosaic screens, using 16 subpictures and a subsample factor of 72 for the base layer. For level three (a), this results in an abstract interval of 196.6 minutes, for level two (b), 12.2 minutes, and for level one (c), 0.8 minute per mosaic screen.

deo navigation form co-exists next to the “conventional” navigation method discussed in Section 3.4, it is beneficial to re-use the video navigation framework presented in Section 3.4.3. The subpictures constructing a mosaic screen are derived from normal-play intraframe-coded pictures. Therefore, the CPI signal processing conducted during recording, which involves bit-stream parsing, is extended with an MPEG-2 scalable decoder and MPEG-2 encoder. For each derived subpicture, two subpicture versions, one with a bounding box and one without, are stored as CPI data in the metadata base. Furthermore, the number of repetition pictures required to transmit a mosaic screen is calculated, which depends on the chosen subpicture bit cost.

C. Signal processing during video navigation

Figure 3.26 shows the flowchart for generating mosaic screens. The mosaic-screen construction starts with determining the applied hierarchical navigation layer, see top of Fig. 3.26. Depending on the selected hierarchical navigation layer, either Equation (3.7), (3.8) or (3.9) is employed for calculating the selected subpictures. For the situation that one or more subpictures are available as reference data in the MPEG-2 decoder, re-use is possible (`re-use == True`?). Slices constructing the re-used subpictures are calculated on the basis of motion-compensated macroblocks. Subpictures that cannot be predicted, are retrieved from the metadata base on the basis of either Equation (3.7), (3.8) or (3.9). Hereby, the calculated values k_1 , k_2 or k_3 form the CPI entry points, relative to the current normal-play playback position. On the basis of solely intra-

Algorithm 10 Mosaic-screen construction using MPEG-coded subpictures**Require:** MPEG-compressed subpictures of fixed bit cost**Ensure:** MPEG-compliant mosaic screen**Initialize:***nr_mini_slices, nr_hor_subpict, nr_vert_subpict*generate sequence header ▷ Seq. header for nav. sequencegenerate sequence extension ▷ Seq. ext. header for nav. sequencegenerate picture header ▷ P-picture picture headergenerate picture extension ▷ P-picture picture coding header

▷ generate MPEG-compliant mosaic screen

for $v = 1$ to *nr_vert_subpict* **do** ▷ vertical subpicture loop **for** $i = 1$ to *nr_mini_slices* **do** ▷ nr. mini slice constructing subpicture **for** $h = 1$ to *nr_hor_subpict* **do** ▷ horizontal subpicture loop select mini slice i from subpicture $v * 4 + h$ ▷ select mini-slice adapt locator information ▷ adjust mini-slice locator information append mini-slice to bit stream ▷ concatenate adjusted mini-slices **end for** **end for****end for**

coded or intra- and inter-coded subpictures, a preliminary mosaic screen is constructed. In order to construct an MPEG-compliant mosaic screen (`Generate final mosaic screen`), a minimum set of MPEG-2 video headers are calculated, followed by adaptation of the `slice_start_code` and `macroblock_address_increment` location information of the mini-slices, constructing this preliminary mosaic screen. We refer to Algorithm 10 for detailed steps. This “still image” is converted into a video navigation sequence by adding frame-based repetition pictures, as indicated in the flowchart of Fig. 3.27. This repetition process equals that of the repetition method deployed for fast-search video navigation, as discussed in Section 3.4.3. Figure 3.26 should be considered as an insert for the block (`Generate mosaic-screen`), see Fig. 3.27. The mosaic-screen sequence generation re-uses parts of the fast-search video sequence generation, as discussed in Section 3.4.3 and depicted at the right-hand side of Fig. 3.27. Due to the fact that a mosaic screen is based on progressive subpictures, the interlaced-related processing is absent. Also absent is the speed-error determination, as the video navigation is controlled by the user. The number of required repetition pictures is calculated during recording and depends on the mosaic-screen bit cost, which is provided by the CPI. Finally, the repetition-picture calculation equals that of the fast-search progressive video situation, as depicted in Fig. 3.13. This flowchart should again be considered as an insert to the block `Generate repetition picture` in Fig. 3.27.

3.7 Experiments and validation results of both navigation concepts

In this section, we propose a PVR block diagram, suitable for implementing the video navigation solutions discussed in Section 3.4 (fast forward/reverse and slow motion) and Section 3.6 (hierarchical navigation). Furthermore, we present experimental and validation results on system aspects of our proposed video navigation solutions.

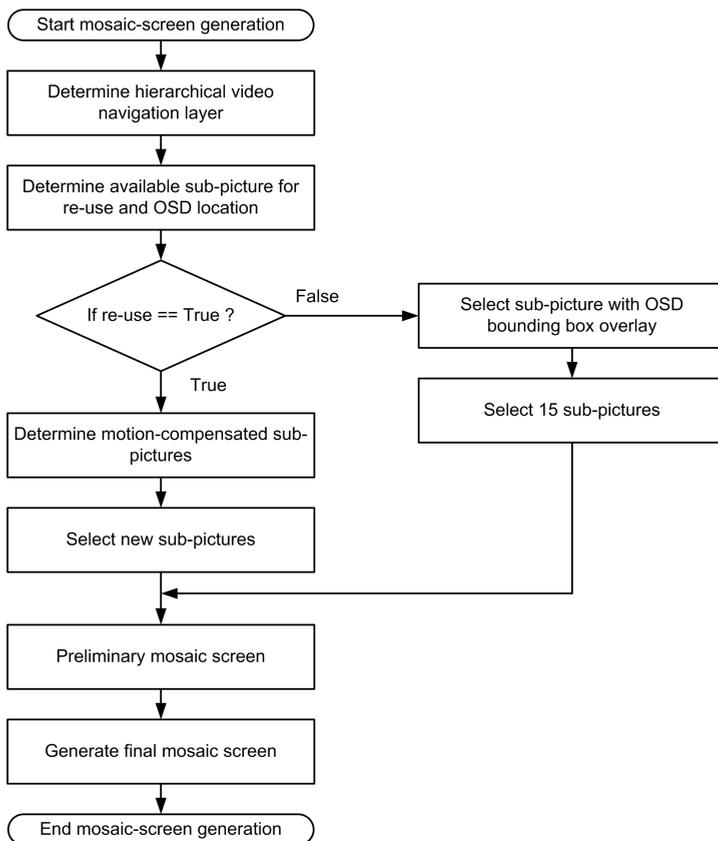


Figure 3.26 — Flowchart of mosaic-screen generation on the basis of re-used MPEG-compressed subpictures.

3.7.1 Functional block diagram of Personal Video Recorder (PVR)

For both solutions, we assume a common architecture of a Personal Video Recorder (PVR), based on a Hard Disk Drive (HDD) capable of storing an MPEG-2 Transport Stream (TS). This system is equipped with functionality supporting full-frame and mosaic-screen-based video navigation. Although both video navigation forms presented in this chapter can be implemented within the boundaries of this PVR block diagram, only fast-search video navigation has been fully implemented and experimentally validated. The validation based on measurements for full-frame slow-motion video navigation could not be completed due to a project abortion. We therefore provide a validation of slow-motion navigation, based on realistic performance estimates, which are derived from the measured fast-search navigation results. Let us first discuss the functional block diagram required for both navigation operations.

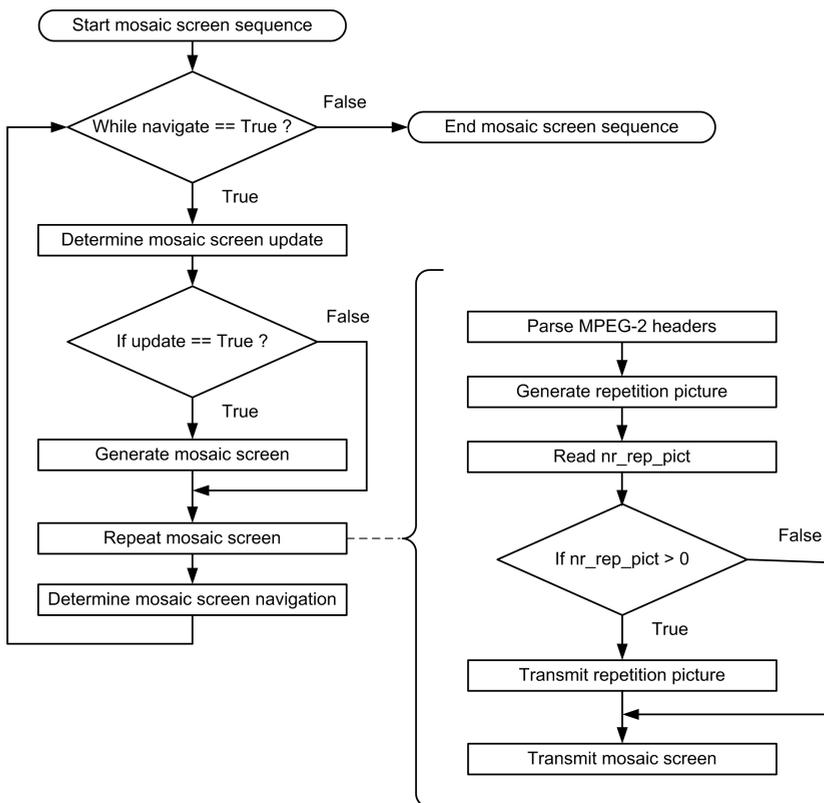


Figure 3.27 — Flowchart for generating mosaic-screen compliant video sequence.

Figure 3.28 portrays a functional block diagram of a networked PVR, where the control part has been omitted. The video navigation processing is separated in two parts: (1) recording and (2) playback. At the left side of the diagram, the compressed MPEG-2 TS enters the storage system. This signal can be viewed in real time, via signal path (a), or in time-shift playback mode via signal path (b). Video navigation playback is available via signal path (c). At the switch output, (at the right side), the TS leaves the PVR system and is supplied to a networked client employing MPEG-2 decoding. During recording, at the left-hand side of Fig. 3.28, Characteristic Point Information (CPI) is generated, extracting parameters as discussed in Section 3.4, which are stored in the “metadata base”. Video navigation playback involves three functional blocks. The first block is “Video navigation processing”, which conducts the specific video navigation playback processing depending on the selected video navigation, as depicted in Fig. 3.29. For fast-search or slow-motion playback, the “Video navigation processing” operates on the provided normal-play TS and associated CPI. Therefore, this block requests the second block “Read-list” for

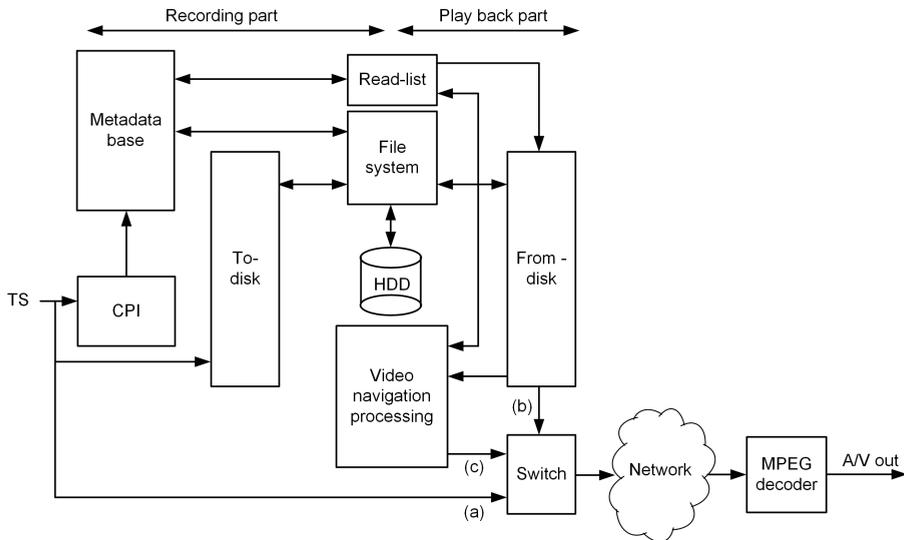


Figure 3.28 — Functional block diagram of a HDD-based PVR. The Switch block selecting signal path (a) indicates the real-time viewing mode; when selecting (b) it operates in time-shift playback; and when selecting (c), the video-navigation operation mode.

playback information signals. The normal-play TS information is retrieved by the third block “From-disk” on the basis of CPI storage locator information, provided by the “Read-list” block. This normal-play TS and the associated CPI is provided to the “Video navigation processing” block and utilized to derive either a fast-search or slow-motion video navigation signal. Note that the read CPI value may be modified in order to influence the video navigation playback. For mosaic-screen video navigation, the “Video navigation processing” block constructs the mosaic screen. Therefore, subpictures are requested from the “Read-list” block, which are retrieved from the “metadata base”, as well as the corresponding CPI data.

We continue by elaborating on the involved navigation signal processing related to the block diagram. Figure 3.29 depicts the processing during video navigation conducted by the “Video navigation processing” block. In order to derive a fast-search or slow-motion video navigation sequence, the normal-play TS is demultiplexed resulting in a video ES, which is parsed to locate the individual pictures.

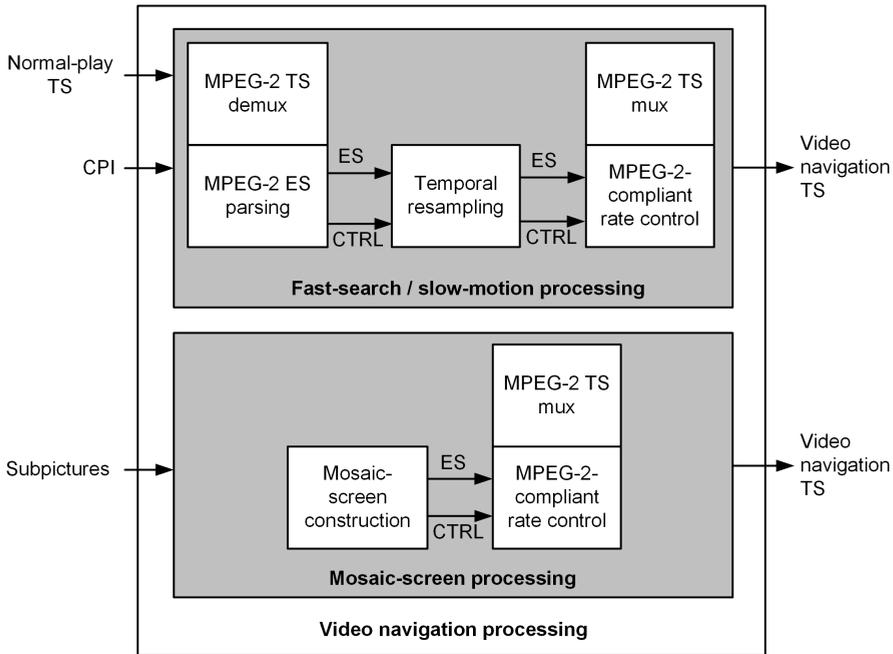


Figure 3.29 — Signal processing flow of video navigation processing for fast-search, slow-motion and hierarchical mosaic-screen video navigation.

For the fast-search playback mode, the “Temporal resampling” block provides the I-pictures to the “MPEG-2-compliant rate control” block. This block adds repetition pictures depending on the provided CPI information. The repetition pictures depend on the video properties determined by the “MPEG-2 ES parsing” block or a CPI-modified version to enforce a particular playback. Finally, the video navigation ES is multiplexed into a video navigation TS.

For slow-motion, the full TS is demultiplexed and the video ES is parsed, determining the properties of each normal-play picture. On the basis of these picture properties, the “MPEG-2-compliant rate control” block inserts either P- or B-type compressed repetition pictures or repeats normal-play B-pictures, after which the obtained slow-motion video navigation sequence is multiplexed into a video navigation TS.

The “Mosaic-screen construction” block creates a mosaic-screen either on the basis of predictive-coded subpictures in combination with inter-coded or intra-coded-only subpictures. This mosaic-screen bit stream is extended with P-type repetition pictures, creating the final video navigation sequence, which is multiplexed into the final video navigation TS.

3.7.2 Performance measurement of implemented fast-search navigation

A full software-based PVR system as described in Section 3.7.1 and equipped with fast-search video navigation as presented in Section 3.4.3, has been realized. The implementation is executed on a mediaprocessor DSP⁵, featuring a VLIW architecture operating as a 5-issue slot machine with a clock frequency of 220 MHz. To analyze the DSP load, a 6 Mbit/s MPEG-2 TS is used as an input. This stream contains both compressed audio and video signals, where the video is coded with a GOP length of $N = 12$. Two use cases are explored.

- *Case 1.* This case is based on fast-search navigation, with a speed-up factor $P_s = 12$ and a picture refresh-rate of 25 Hz, which equals the nominal European television frame rate.
- *Case 2.* This case refers to fast-search navigation with a speed-up factor $P_s = 4$. This factor results in a picture refresh-rate of 8.33 Hz.

Table 3.5 — *Measured DSP clock-frequency fraction required for fast-search navigation (normal-play GOP length $N = 12$). The measured is a fraction of the total execution clock frequency measured with a dedicated tool and rounded to an integer number.*

Speed-up factor	Picture refresh-rate (Hz)	DSP load (MHz)	DSP load (%)
12	25	33	15.0
4	8.33	12	5.5

3.7.3 Performance measurement of implemented fast-search navigation

Table 3.5 indicates the measured DSP load for both use cases. In Table 3.5, the DSP load is expressed in terms of the required clock frequency (MHz) for real-time execution, which also includes the execution parallelism factor in the mediaprocessor DSP. Row 1 indicates the situation where every normal-play I-picture is retrieved from disk and used for constructing the fast-search navigation sequence, with a nominal refresh-rate. The measured DSP load indicates the video navigation signal processing involving demultiplexing, ES parsing, MPEG-compliant rate control and final TS multiplexing, as depicted at the top of Fig. 3.29. In Row 2, again each I-picture is recovered, but due to the lower playback speed-up factor, the rate control inserts two repetition pictures, resulting in a lower refresh-rate emulating the indicated fast-search speed-up factor $P_s = 4$. This reduces the amount of retrieved I-pictures per second with a factor of three, thereby requiring less processing (approximately one third).

From this experiment, we conclude that the execution of the involved signal processing scales linearly with the offered normal-play GOP size and the chosen speed-up factor. This is also plausible, since the involved processing executed on the mediaprocessor is based on a streaming architecture. This conclusion is further supported by the observation that successive I-pictures from consecutive GOPs have approximately the same bit cost. Figure 3.30 presents the visual impact of the fast-search navigation for the above-mentioned cases of searching. Figure 3.30(a) indicates a successive normal-play I-picture with different contents, due to the significant differences in temporal positions. Figure 3.30(b) reveals the impact of a tripled bit cost of the selected I-picture, e.g. due to a higher picture quality or more complex data. Due to this high bit cost, the involved picture transmission time triples accordingly, which requires smoothing to come to the desired average bit rate. Consequently, the involved

⁵Commercially available as part of a set-top box chip PNX8525 of Philips Semiconductors.



Figure 3.30 — Navigation snapshots covering 240 ms of consecutive frames. The figure should be column-wise interpreted. (a) Selected normal-play I-pictures for GOP length $N = 12$ and refresh-rate of 25 Hz, $P_s = 12$. (b) Selected normal-play I-pictures for GOP length $N = 12$ and refresh-rate of 8.33 Hz, $P_s = 12$. (c) Selected normal-play I-pictures for GOP length $N = 12$ and refresh-rate of 8.33 Hz, $P_s = 4$.

bit-cost smoothing requires three frame periods, leading to a twofold picture repetition. This leads to rendering the same picture three times. In order to avoid a navigation playback speed-error, we need to skip two successive I-

pictures. Figure 3.30(c) shows the visual impact when applying repetition pictures and not skipping successive I-pictures. This results in a threefold lower navigation playback speed. A disadvantage of the last two methods is that the original motion is not preserved due to the repetition process.

For fast-search navigation experiments, we conclude with two findings regarding system aspects for this type of navigation.

- *HDD throughput and system bandwidth.* Lowering of the refresh-rate can be exploited to significantly reduce the DSP or CPU load. We have found that a reduced refresh-rate is a concept that can be used to solve various problems: (1) to overcome a possible HDD seek-rate constraint, (2) to reduce the load on the system bandwidth and (3) adjust the Quality-of-Service (QoS).
- *Perceptual aspects of the navigation.* A reduced refresh-rate can also be used to accommodate the visual perception experienced by the viewer during high-speed fast search. Due to little or no temporal correlation, the visual information during high-speed playback can hardly be followed by the viewer. Lowering the refresh-rate causes the viewer to better perceive the video information. However, when applied during high-speed search, e.g. $P_s = 100$ while preserving the speed-up factor, the navigation efficiency declines, as a considerable amount of normal-play information is disregarded. During our fast-search video navigation experiments, we have found that the normalized refresh-rate should not be lower than $1/3$, as indicated by a gray area in Fig. 3.31. This is because a lower refresh-rate results in a slide-show effect, where the viewer loses a fast-search viewing experience.

3.7.4 Performance estimation of slow-motion navigation

Similar to fast-search navigation, slow-motion navigation is obtained on the basis of re-used normal-play MPEG-compressed video information. Evidently, the involved slow-motion signal processing has a high commonality with the fast-search processing. It was shown in the fast-search experiments that the measured execution fraction performance on the DSP grows linearly with the frame rate. Due to both observations and the conceptual similarity, we have omitted this extra implementation effort and did not repeat the same experiment for slow-motion navigation. Instead, we conduct a performance estimation on the basis of the measured fast-search performance.

Let us now derive how the fast-search numbers can be translated into the slow-motion case, where the emphasis is on calculating the slow-motion DSP clock

cycle performance. The normal-play video sequence deployed in the fast-search performance measurement has been encoded at 6 Mbit/s, employing a GOP length $N = 12$. The I-pictures have been encoded such that the bit cost is maximally 600 kbits. When using this maximum for re-used I-pictures and a maximum refresh-rate of 25 Hz, we obtain a maximum bit rate for navigation of 15 Mbit/s. For this maximum throughput rate, and the measured clock cycle fraction, we can now define a linear function, revealing the relation between the DSP load as function of the processing bit rate, resulting in $33/15 = 2.2$ clock cycles/bit. This leads to a linear function $y = 2.2x$, with y being the DSP load in clock cycles and x the input bit rate in Mbit/s. Based on this linear function, we can now estimate the performance load for slow-motion video navigation. Table 3.6 indicates the results of this estimation function in the form of the DSP estimated clock-cycle loads for three slow-motion playback speeds. The performance estimates depicted in Table 3.6, show a declined clock-cycle consumption for lower playback speeds. This is due to the fact that for lower playback speeds, each normal-play picture is several times repeated, resulting in a reduced bit-stream parsing per time instance.

From the estimated DSP clock frequency fraction, we conclude that slow-motion video navigation based on re-used MPEG-compressed normal-play video information results in a quite low computation load compared to the fast-search computation load.

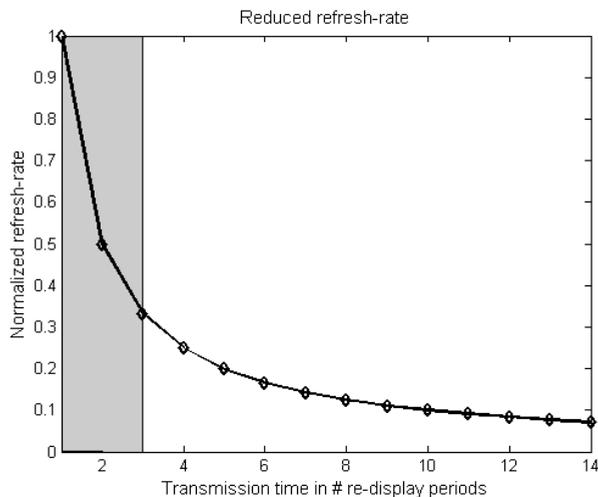


Figure 3.31 — *Reduced refresh-rate.* The gray region at the left indicates the preferred region of operation for fast-search navigation.

Table 3.6 — *Estimated DSP clock frequency fraction required for slow-motion navigation (normal-play bit rate 6 Mbit/s). The estimated clock frequency is rounded to an integer number with a ceiling function.*

Slow-motion factor	Input bit rate (Mbit/s)	DSP load (MHz)	DSP load (%)
1/2	3	7	3.2
1/3	2	5	2.3
1/4	1.5	4	1.8

Table 3.7 — *Estimated DSP clock frequency required for deriving the CPI metadata during recording. The estimated clock frequency is a fraction of the processor clock frequency rounded to an integer number.*

Input bit rate (Mbit/s)	DSP load (MHz)	DSP load (%)
15	17	7.7
9	10	4.5
4	5	2.3

3.7.5 Performance estimation of navigation processing during recording

For flawless video navigation playback, a preparation signal processing action is already performed during recording of the scene. This involves MPEG-2 demultiplexing and Characteristic Point Information (CPI) bit-stream parsing, in order to determine the GOP structure, especially the length N and P-distance M . These parameters are additionally stored as metadata with the recording, so that they are readily available when navigation starts. In this way, dual-stream processing is avoided and also the associated clock-cycle consumption. Similar to slow-motion, we estimate the required performance for the previous operations at recording. The video navigation signal processing during recording involves TS demultiplexing and video ES parsing in order to derive the CPI. These two operations are also applied during video navigation playback and form half of the involved signal processing chain. Based on this observation and the derived linear function from the previous subsection on slow-motion, we estimate the clock-cycle performance during recording. The involved processing for deriving the CPI at recording is estimated from Fig. 3.29. In this

figure, the fast-search processing is depicted at the top of the figure and consists of demultiplexing followed by ES parsing. The remaining processing is not conducted when deriving the CPI during recording. As a result, the computations involved with the CPI processing are approximately halved. As a conclusion, the linear function between the throughput rate (parameter x) and the resulting performance in clock cycles (parameter y) is also halved, leading to a function $y = 1.1x$. Table 3.7 indicates the DSP estimated clock-cycle load for three normal-play processing bit rates.

On the basis of the estimated DSP clock frequency, we conclude that the video navigation processing conducted during recording, i.e. the CPI metadata generation, is estimated to also require a low computation load.

3.7.6 Picture quality validation of mosaic screens

The mosaic-screen quality is directly related to the employed bit cost for coding of the individual subpictures. These subpictures constructing a mosaic screen are derived during recording and stored as metadata, on the basis of normal-play intraframe-coded pictures. Figure 3.32 indicates the processing chain for deriving two subpictures, one with OSD and one without. Hereby, the normal-play MPEG-2 TS is demultiplexed, followed by a scalable MPEG-2 decoder, operating only on normal-play intraframe-coded pictures. The encoding of the two subpictures is implemented such that the video part of the subpictures is shared, enabling a constant picture quality. In this way, the picture quality between an OSD-equipped subpicture equals that of its counterpart, avoiding a visible change in picture quality when conducting intra mosaic-screen navigation. Let us now determine the quality level for individual subpictures of the mosaic screen. In Section 3.6.2, we have discussed that the “mini-slices” within the subpictures are coded with a fixed bit cost. To determine this bit cost, we have conducted a number of perceptual quality evaluations. The subpictures should be of good quality because they are used for navigation purposes and

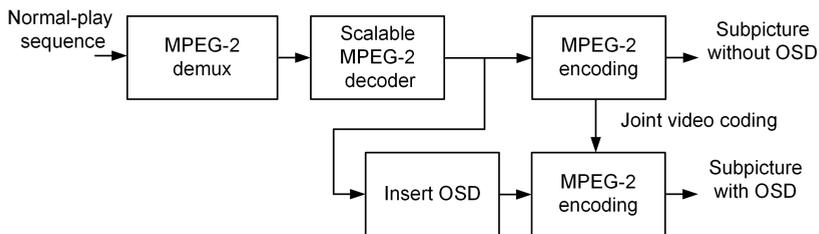


Figure 3.32 — *Subpicture processing chain for mosaic screens.*

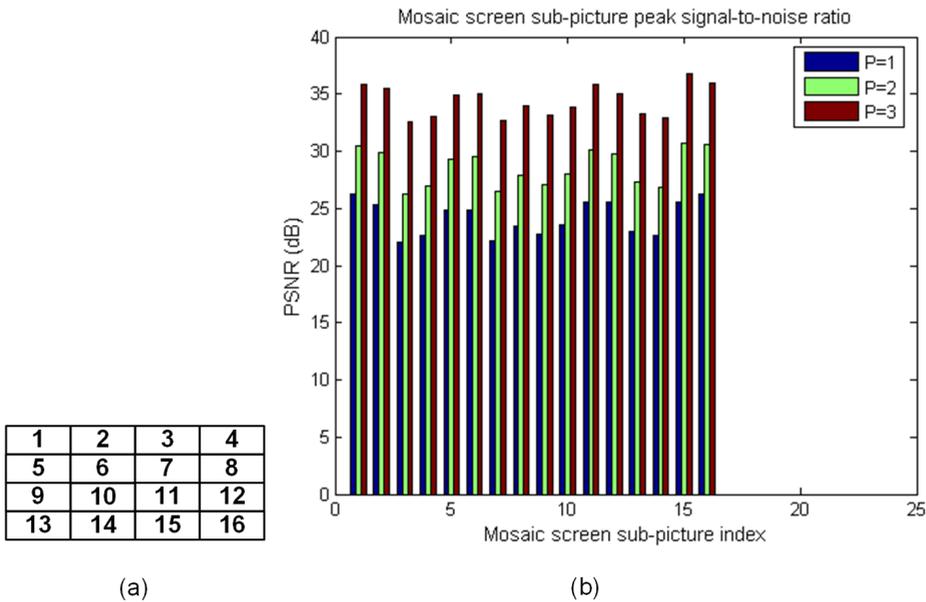


Figure 3.33 — PSNR of the subpictures of a mosaic screen. (a) Subpicture numbering of the mosaic screen. (b) PSNR of per subpicture of the mosaic screen, corresponding with a transmission time p of 1, 2 or 3 frame periods and a transmission rate of 15 Mbit/s.

may therefore be visually inspected by the viewer. The quality choice is emphasized for the situation that a subpicture is tracked by the eye of the viewer during e.g. a scrolling operation. For the subjective experiments, a modified version of the MPEG-2 reference encoder software [92] was used. The primary modifications deal with the creation of mini-slices and the involved bit-rate control for fixed-cost mini-slices. The modified MPEG-2 reference encoder software [92], which employs a single-pass encoder, delivers a bit cost that is lower than the maximum bit cost indicated in Table 3.8. The fixed bit cost per mini-slice is obtained on the basis of padding, employing zero-valued Bytes.

In our experiments, we have used the parameter setting of a standard MPEG-2 MP@ML encoder setting, facilitating a bit rate of 15 Mbit/s. Using the 25-Hz frame rate for European broadcasting, the video coder running at 15 Mbit/s, fills the VBV-buffer in almost three frame periods. For our subjective experiments, we have encoded a mosaic screen at three different bit costs. The mosaic-screen bit cost equals the product of the maximum MPEG-2 ML bit

rate, the European frame period and a scale factor p , which denotes the mosaic-screen transmission time expressed in frame periods. Figure 3.34 visualizes the subpicture quality for the three different mosaic-screen bit costs, which require a transmission time as indicated in Table 3.8. It can be observed that the perceptual quality increases with increased subpicture bit cost. Because a mosaic screen consists of multiple subpictures, the total picture quality is determined by the quality of each individual subpicture. Figure 3.33(b) depicts the Peak Signal-to-Noise Ratio (PSNR) of each individual subpicture that is encoded using the single-pass MPEG encoder. A good mosaic-screen picture quality can be expected when individual subpictures have a PSNR of 30 dB or more. On the basis of the previous subjective and objective evidence, we conclude that a mosaic screen constructed from subpictures with PSNR > 30 dB, requires a bit-cost setting of 225,000 Bytes per mosaic screen. This bit cost is almost the complete VBV buffer size, which equals 1,835,008 bits (224 kBytes) for MPEG-2 MP@ML. Let us now continue with the visual inspection of an example of a hierarchical mosaic screen. Figure 3.35 provides an instant hierarchical overview at three different time scales of the same video sequence, according to the algorithm of Section 3.6.3.

Hereby, the base-layer mosaic screen provides a detailed normal-play instant overview, which is stretched to more visible samples of individual scenes from the sequence, eventually ending into equidistant snapshots of single pictures constructing the video sequence. Fig. 3.35 reveals the visual relation between the hierarchical navigation layers, where a higher layer mosaic screen is constructed from 16 adjacent mosaic screens located at a lower layer. This becomes clear when visually inspecting the first mosaic screen at the second hierarchical layer (middle). This mosaic screen is constructed using subpictures derived from the base layer (bottom). In this example, the first three base-layer mosaic screens all provide a subpicture, located at the upper-left corner, to con-

Table 3.8 — *Bit cost for mosaic screens in case of various transmission times at a bit rate of 15 Mbit/s.*

Mosaic-screen bit cost (Bytes)	Transm. time p (No. frame periods)	Mosaic-screen transm. time (ms)	Subjective quality
75,000	1	40	poor
150,000	2	80	modest
225,000	3	120	good

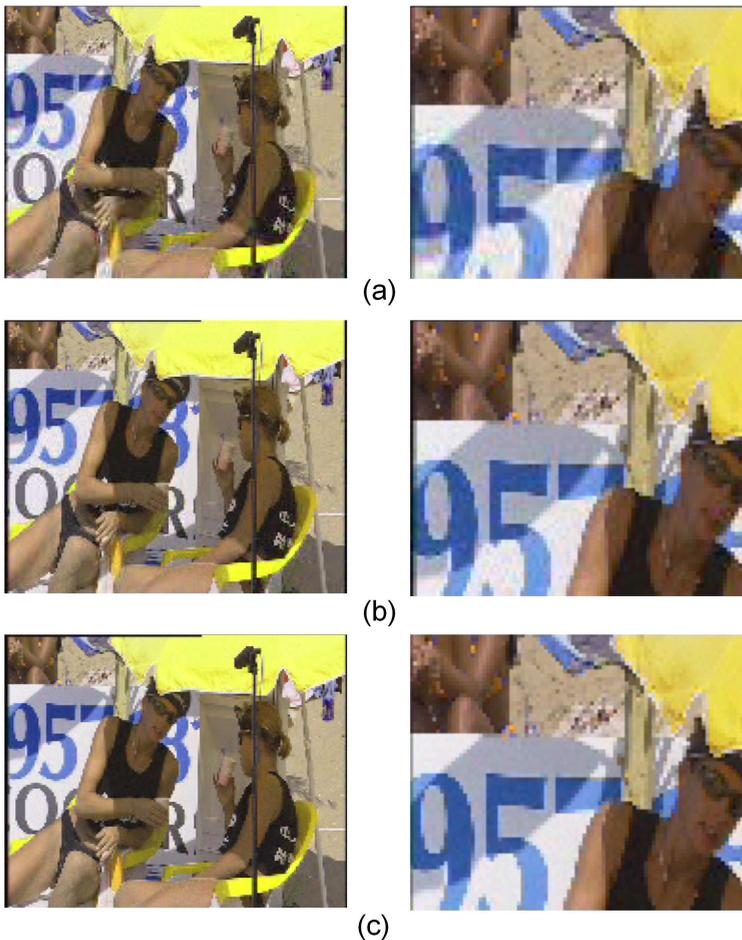


Figure 3.34 — Subpicture quality using fixed bit cost for the “mini-slices”. At the left column, full pictures are shown, while at the right column a magnified view of the left picture is shown. (a) Subpicture quality for mosaic screen with a bit cost of 75,000 Bytes. (b) Subpicture quality for mosaic screen with a bit cost of 150,000 Bytes. (c) Subpicture quality for mosaic screen with a bit cost of 225,000 Bytes.

struct the first mosaic screen at the second hierarchical navigation layer. A similar situation applies to the mosaic screens located at the third hierarchical layer (upper), which are based on mosaic screens from the second hierarchical navigation layer.

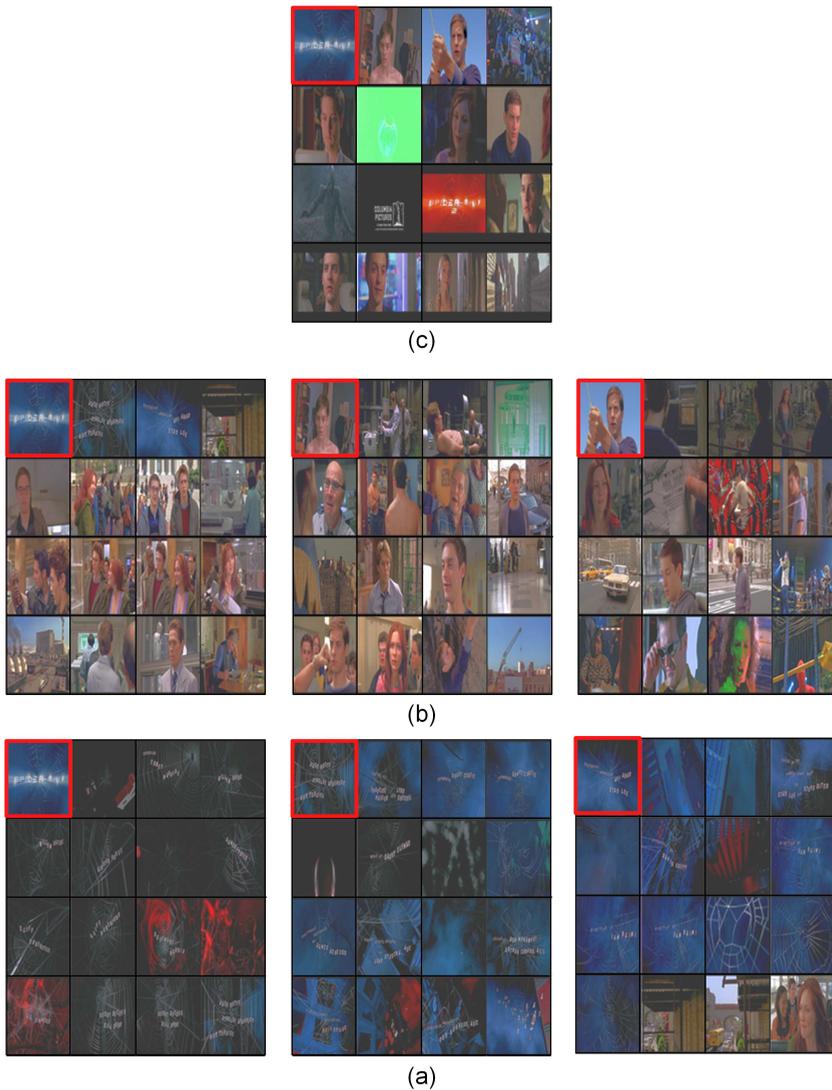


Figure 3.35 — Example of hierarchical video navigation. (a) Base layer providing an instant overview of 0.8 minutes. (b) Second layer providing an instant overview of 12.2 minutes. (c) Third layer providing an instant overview of 196.6 minutes.

3.8 Discussion on automated mosaic screen scrolling

Navigation on the basis of mosaic screens as discussed in this chapter, requires user interaction in order to request a new mosaic screen. Navigation through mosaic screens corresponding to different time scales (different hierarchical layers) is achieved in a convenient manner, due to the limited number of hierarchical layers. Scrolling (browsing) through mosaic screens belonging to a single hierarchical layer may become a tedious task, especially when conducted at the lowest hierarchical layer, i.e. the base layer. The root cause for this is the manual interaction involved with the storage system, to request for a new mosaic screen.

In order to avoid this, we propose here an automated scrolling technique, which is a concept to avoid individual requests resulting in new mosaic screens, so that the involved interaction is minimized. Based on a smooth horizontal motion portrayal, subpictures constructing a neighboring mosaic screen enter the display (shift in) automatically, while from the available visible mosaic screens, the subpictures disappear from the display (shift out) in opposite direction. The shifting direction of mosaic screens on the display is controlled by the indicated searching direction from the viewer. Such an approach offers a highly intuitive user interface with a natural understanding for navigation.

However, the implementation of the above novel concept contains some challenges. The key issue is the available throughput rate of the adopted execution architecture, in our case the CPU-DSP combination with the streaming operations at the DSP. The proposed user interface requires continuous decoding, screen composition, shifting and final encoding, which forms a considerable load on the DSP throughput and CPU. A possible solution circumventing this high throughput and CPU load, is the usage of the MPEG decoder for motion-based shifting of the mosaic screen, utilizing the available infrastructure for motion compensation. Motion compensation is a coding feature available when employing predictive coding and is supported when deploying P-type coding syntax. This coding feature was already proposed for the coding of mini slices constructing the subpicture, allowing the re-use of subpictures already available at the MPEG decoder as reference information. Automated scrolling through neighboring mosaic screens is a navigation feature, which typically results in a high correlation between successive mosaic screens. This high amount of correlation allows the re-use of predictive coding, but now for mosaic screens, which would reduce the amount of new video information and thus throughput rate, while simultaneously reducing the involved CPU load. For a practical implementation, motion compensation can be limited to an integer multiple of 16 pixels.

3.9 Conclusions

In this chapter, we have classified video navigation into three categories, of which two categories are characterized by a networked decoder, operating at a distance from a storage device elsewhere in the network. These two categories have been elaborated and worked out experimentally in this chapter. The third category involves a different navigation concept, which is not networked and requires local audiovisual decoding. The two proposed video navigation categories from this chapter are suitable to be employed in a client-server-based communication system. Both solutions feature communication interoperability, based on standard MPEG encoding and decoding techniques and coded MPEG information transmitted across the network. In this way, we circumvent the usage of non-mandatory communication options provided by MPEG, so that the operation of our intended navigation is ensured under all conditions. Let us now briefly discuss the most important findings of our two proposed navigation concepts elaborated in this chapter.

Networked full-frame video navigation. Video navigation on the basis of this concept fully re-uses intraframe MPEG-compressed normal-play video pictures for deriving fast-search navigation. The implementation is based on the finding of characteristic point information during recording, revealing the storage location of these intra-coded pictures, which are then re-used for generation of a fast-search navigation sequence. In a similar fashion, we have implemented slow-motion video navigation on the basis of full re-use of all normal-play pictures. In order to reduce the playback speed, we employ repetition pictures, which repeat normal-play reference pictures. The use of repetition pictures is elegantly employed to also control the bit rate and frame rate of the video navigation sequence. This is achieved by assuming a fixed bit rate and then calculating the transmission time for each re-used intra-coded picture. In a similar way, for both video navigation playback speeds, we control the rendering at field or frame level, depending on the video format (interlaced or progressive). In this way, we efficiently remove field-based video information (interlace kill), thereby avoiding motion judder during navigation.

Networked hierarchical mosaic-screen video navigation. Video navigation on the basis of hierarchical mosaic screens relies on the usage of subpictures derived from intra-coded normal-play pictures during recording. A set of consecutive subpictures construct a mosaic screen. Flexibility in the composition of mosaic screens is essential for providing different temporal instant overviews, covering short-, medium- or long-time intervals. This flexibility is obtained by coding each subpicture at a fixed bit cost, thereby enabling easy retrieval from the storage device while simplifying the construction of the final mosaic screen with the compressed subpictures. A fixed-cost subpicture is achieved by dividing

each subpicture into a set of “*mini slices*”, which are also encoded at a fixed bit cost. Furthermore, when encoding subpictures using P-type coding syntax, new mosaic screens can be constructed using predictive coding, based on re-using subpictures available at the MPEG decoder, thereby reducing the involved throughput and CPU and DSP load.

Let us now summarize the results of the experiments for both the full-frame video navigation solution and the hierarchical mosaic-screen approach.

We have found that the use of repetition pictures in a full-frame video navigation system facilitates various forms of control, which allow to influence the Quality of Experience (QoE) during navigation. When employing repetition pictures to derive video playback with medium playback speed $P_s = 4$, the QoE suffers due to the lack of motion portrayal. However, at high video playback speed $P_s = 50$, the QoE is improved due to an increased display time. Furthermore, when employing *interlace kill* to repeat a picture with interlaced format, motion judder is avoided, which significantly contributes to the perceived QoE. Moreover, the usage of repetition pictures reduces the involved CPU and DSP load. The usage of repetition pictures to reduce the refresh-rate is bounded to $1/3$, as for lower refresh-rates, a slide-show effect occurs, whereby the viewer loses the fast-search navigation experience. Finally, due to the full re-use of normal-play encoded pictures, there is no difference in spatial picture quality between normal play and video navigation.

Experiments with the method employing video navigation with hierarchical mosaic screens, show that this approach provides an attractive instant overview of a normal-play video segment on the basis of equidistant temporal subsampling. In this way, the video navigation inefficiency with respect to full-picture rendering is eliminated, which occurs for video navigation at high playback speeds $P_s > 25$, due to the picture refresh-rate of individual pictures. Furthermore, when increasing the temporal subsampling for higher navigation layers, the video information is condensed into a limited set of mosaic screens. The picture quality of a mosaic screen is determined by the quality of the individual subpictures. When employing a bit cost for a mosaic screen which equals almost the full VBV buffer size of the MPEG decoder, a good picture quality is obtained. We have found empirically that the PSNR of subpictures should be about 30 dB or higher.

Since the video navigation solution is embedded in the overall system structure of a networked storage system, multiple system aspects play a role. We conclude here on the most important aspects. The full-frame video navigation method involves analysis of the recorded video ES, in order to determine various essential stream aspects, which are required at video navigation playback. Essential elements are the normal-play GOP length, the P-distance and the storage location of the intraframe-compressed picture and its size. These param-

ters enable video navigation without speed-errors and efficient retrieval of the re-used pictures and their transmission times.

An example of the complexity analysis leads to the following numbers. When separating the video navigation processing in two parts, i.e. during recording and video navigation playback, the involved DSP load is kept at an acceptable level. For a typical video broadcast at SD resolution and 4 Mbit/s, the DSP load during recording requires a cycle load of 5 MHz, while a DSP cycle load of 22 MHz is required for full frame-rate fast-search video playback with speed-up factor 12 and 5 MHz for slow-motion with playback speed 0.5. For fast-search video navigation, a reduced refresh-rate also decreases the involved DSP cycle load. This also applies to slow-motion playback mode, where an increased slow-motion factor reduces the DSP cycle load.

With respect to the computational load of video navigation with hierarchical mosaic screens, there is an important aspect related to the generation of subpictures. The mosaic-screen technique involves the continuous derivation of subpictures from the received intra-coded normal-play pictures. When considering a typical video broadcast signal, these I-pictures appear at a rate of 2 Hz, leading to the generation of 2 subpictures per second. The derivation of subpictures is quite computationally intensive, involving MPEG-2 demultiplexing and scalable decoding followed by fixed bit-cost encoding of the derived subpicture. In order to lower the throughput and processing load for deriving the subpictures, a single subpicture is selected per normal-play scene. When considering a typical scene duration of 3 seconds, this processing load is reduced to 0.3 Hz, leading to 1 subpicture per 3 seconds. The construction of a mosaic screen occurs in the compressed domain involving bit-stream parsing, rate control on the basis of repetition pictures and MPEG-2 TS multiplexing. This processing shows a high resemblance with the processing required for fast-search full-frame video navigation. We therefore expect that the involved playback processing for mosaic-screen navigation will show a similar throughput and DSP cycle load.

We have presented two forms of video navigation for client-server-based navigation with a low complexity, so that the method can be embedded in the existing infrastructure. By separating the involved signal processing into a recording and video navigation stage, we have been able to divide computational complexity into two more or less equal parts. In this approach, the signal processing during recording prepares information, obtained on the basis of normal-play GOPs, required by the signal processing for video navigation playback, thereby avoiding dual-stream parsing during navigation playback. This benefit holds for both navigation approaches. Another benefit of the embedded nature of the proposed techniques is that the methods can be upgraded over time, facilitating more advanced video navigation features.

A drawback of the proposed video navigation solutions is the usage of a

single information source only (pictorial data), which limits the perceived information by the viewer. For example, auditive information maybe useful for video navigation as well.

Networked video navigation employing hierarchical mosaic screens involves a manual interaction for navigation through mosaic screens, belonging to the same hierarchical navigation layer. In order to avoid manual interaction, automated scrolling is proposed, employing motion-compensated mosaic-screen construction, in combination with intraframe-compressed subpicture fragments. In this way, mosaic screens are shifted in either left or right direction, facilitating horizontal scrolling in both directions.

The presented two video navigation solutions based on a client-server approach utilize only video information, which limits the instantaneously received information. In the next chapter, we will propose a video navigation solution, which employs also the audio information associated with the video information. In this way, both visual and auditory cues are employed, enabling the viewer to simultaneously conduct other tasks, thereby eliminating the need to have eye contact with the display.

“Better to remain silent and be thought a fool than to speak out and remove all doubt.”
Abraham Lincoln, 1809 – 1865

Audio-enhanced dual-window video navigation

4.1 Introduction

In Chapter 3 we have presented three video navigation use cases and proposed a specific navigation solution for both short-time and long-time interval use cases. The proposed specific solutions address two important system aspects, addressing also efficiency: (1) network-based interoperability and (2) decoupling the individual picture rendering rate from the employed video frame rate. In this chapter, we present a solution for medium-time interval navigation, which is the third video navigation use case. Although the proposed video navigation solutions for the short-time and long-time interval can be also employed as a solution for the medium-time interval use case, there is a limitation in the navigation information, as both solutions employ only video signals. This limits the perceived information by the viewer, making these navigation methods less suitable as a *viewing* mode to obtain a summarization (global overview) of



Figure 4.1 — *Dual-window video rendering for medium-time navigation. The main window shows normal-play fragments with the associated audio also being rendered, whereas the PiP window shows video corresponding to the fast search.*

the stored audiovisual information. Moreover, the short-time and long-time interval navigation solutions require the viewer to maintain visual contact with the display, limiting the viewer's freedom to conduct other tasks in parallel with navigation.

In order to establish a navigation form suitable for visual summarization, while simultaneously offering user freedom in performing other tasks, we propose a medium-time interval video navigation, with an improved summarization and additional auditive information, which is presented to the viewer. The video navigation through summarization is improved by detailing video sequence information and combining this with fast-search video information, which is presented in a dual-window fashion.

More specifically, we propose a novel audio-enhanced dual-window navigation, based on a conventional primary image with associated audio, combined with an additional Picture-in-Picture (PiP) screen, as depicted in Fig. 4.1. Hereby, the primary window displays fragments of normal-play video, while conventional video trick-mode playback, such as fast forward or fast reverse, is presented in the PiP window. In this chapter, we mean with audio-enhanced navigation that the navigation is supported with audio signals (not that the audio itself is enhanced).

The usage of audiovisual information for navigation purposes has multiple advantages compared to a video-only navigation solution. A first advantage is that the audio information forms an additional information source, making the combined audiovisual navigation information more informative. A second advantage is that the presence of auditive information avoids the need for a viewer to have permanent visual contact with the display, thereby enabling the viewer to perform other activities in parallel, while the viewer becomes less fatigued or bored from the navigation.

A navigation solution based on audiovisual information, combined with fast-search video information differs conceptually from a video-only navigation method, in the sense that both audio and video information signals should be simultaneously handled by the storage system, resulting in additional system requirements. These system requirements are further expanded due to the difference between the perception of audio and video information. As the medium-time interval video navigation solution will coexist next to the short-time and long-time interval video navigation solutions, this enables the sharing of functional parts of these navigation solutions. However, the medium-time navigation employs dual-signal video information, which cannot be received by a standard DTV platform. In order to avoid the transmission of a dual-signal video component in the network, both video signals need to be reformatted into a single video sequence. This involves video mixing and re-encoding and subsequent multiplexing of the audio signal data, which is highly complicating the implementation in the basic DTV architecture. To simplify this issue, we aim at a concept employing only local playback at the actual platform of this

dual-window navigation solution, avoiding the need for full re-encoding of the video navigation signal.

Our proposed navigation solution involves audio information and two independent video signals, which requires a secondary video decoder. Modern high-end consumer platforms offer such a secondary video decoder, whereas in a typical low-cost consumer platform, such a secondary video decoder is absent. For this situation, additional video decoding on a general-purpose control processor forms an attractive solution. As video decoding is a computationally intensive task, we aim at employing complexity-scalable video decoding, enabling the smooth operation of the CPU tasks. Complexity-scalable decoding algorithms exist for MPEG-2 intraframe-coded pictures. Although a scalable video decoding algorithm for H.264/MPEG4-AVC intraframe-coded pictures exists, this solution suffers from drift [93]. For thumbnail-sized pictures, this drift may be acceptable, but for high-resolution pictures this distortion is not acceptable.

Summarizing, this chapter aims at developing a medium-time interval navigation solution, where dual-window video is employed for normal-play and fast-search video information, enhanced with auditive information. The solution should be deployed on a typical low-cost consumer DTV platform. Furthermore, we propose a low-cost drift-free decoding method for H.264/MPEG4-AVC intraframe-coded images, to derive the fast-search PiP-sized pictures.

The remainder of this chapter is as follows. First, we present background information on related work and system requirements in Section 4.2. Section 4.3 elaborates on our conceptual solution for audio-enhanced dual-window video navigation. System integration and implementation are discussed in Section 4.4, while our proposal for computational reduced H.264/MPEG4-AVC intraframe decoding is presented in Section 4.5. Section 4.6 discusses experimental results and conclusions are presented in Section 4.7.

4.2 Background and system aspects

This section briefly elaborates on the perception of auditive information for navigation purposes. Moreover, it discusses scalable decoding results from literature for both MPEG-2 and H.264/MPEG4-AVC intraframe-coded pictures. Finally, audiovisual signal processing aspects relevant for system integration are addressed.

A. Consecutive sound bursts for interpretation of audio information

The usage of audio information for video navigation has been limited to either playback speeds around unity [14], or for feature extraction, enabling advanced intra-program video navigation [75]. A possible solution direction for

advanced intra-program video navigation is the simultaneous rendering of multiple information signals. Since human perception employs both visual and auditory cues, it is opportune to employ audio information associated with the video-navigation information, thereby creating a multi-source information signal for advanced intra-program video navigation. However, cognitive processing of sound occurs at multiple time scales. In audio perception, these time scales range from microseconds for localization of sound sources, via milliseconds for pitch analysis and event detection, to tens of milliseconds for analyzing speech characteristics. In mobile communication, phone ringing recognition requires about 100 milliseconds, while syllables and words in speech require larger time scales [94]. As a result of this, it is not effective to select audio information associated with a single picture forming the video navigation signal, which generates 40/33 milliseconds of consecutive sound for a 25/30-Hz television system.

B. Complexity-scalable video decoding

In general, the concept of complexity-scalable video decoding may form an attractive solution to control decoding complexity and thus the required system resources [88]. In order to reduce the spatial resolution, the scalable decoding technique operates in the transform domain, where a sub-set of the transform coefficients are used to reconstruct a set of pixels, forming a reduced block size, compared to the original encoding block size. Typical spatial reduction factors for an 8×8 coefficient block, while preserving the aspect ratio, are horizontally and vertically a factor of 8, 4 or 2. A fast-search navigation signal with a PiP

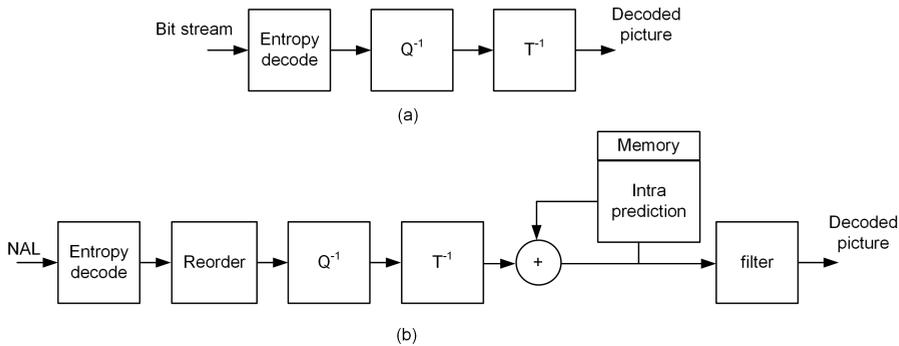


Figure 4.2 — Basic block diagrams for intraframe decoding systems. (a) MPEG-2. (b) H.264/MPEG4-AVC.

size on the display, represents a video signal with inherently reduced spatial resolution, originating from intraframe-coded pictures. Therefore, this video PiP signal is suitable to be obtained via computational reduced decoding techniques. Hereby, a distinction is made between MPEG-2 and H.264/MPEG4-AVC compressed video sequences. Figure 4.2 indicates the basic block diagrams for an MPEG-2 and H.264/MPEG4-AVC intraframe video decoder. The main difference between the two intraframe coding schemes is that for MPEG-2 after the inverse transformation, block-based pixel data is available, whereas for H.264/MPEG4-AVC, the result after inverse transformation is a block-based residual signal. This requires the local reconstruction of the spatial block-based predictor for reconstructing the final pixels. A complexity-scalable compression technique for MPEG-2 is described in [88], while for scalable decoding a solution is presented in [91] and for H.264/MPEG4-AVC in [95][93].

C. Scalable MPEG-2 intraframe decoder

In Section 3.6.2, scalable MPEG-2 intraframe decoding has been elaborated in depth. In this subsection, a brief summary is presented on the basic concept of scalable MPEG-2 decoding. In MPEG-2 video decorrelation is achieved by means of an 8×8 2D-DCT, which enables a scalable decoder to reconstruct a spatial region that has fewer pixels. Figure 4.3 indicates such a form of spatial scalability, where a selection of four coefficients (upper-left corner) from an 8×8 DCT matrix results in a spatial region size of 2×2 pixels, effectively reducing the original 8×8 region by a factor four in horizontal and vertical direction.

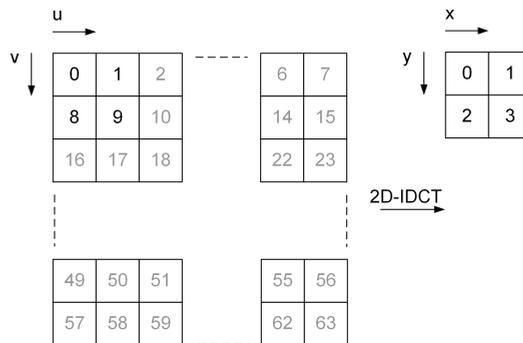


Figure 4.3 — Scalable 2×2 IDCT for an SD-to-QCIF MPEG-2 decoder. At the left, the received 8×8 coefficients (0..63) are depicted. At the right, a 2×2 pixel block is depicted, which is obtained by applying a 2×2 scalable IDCT.

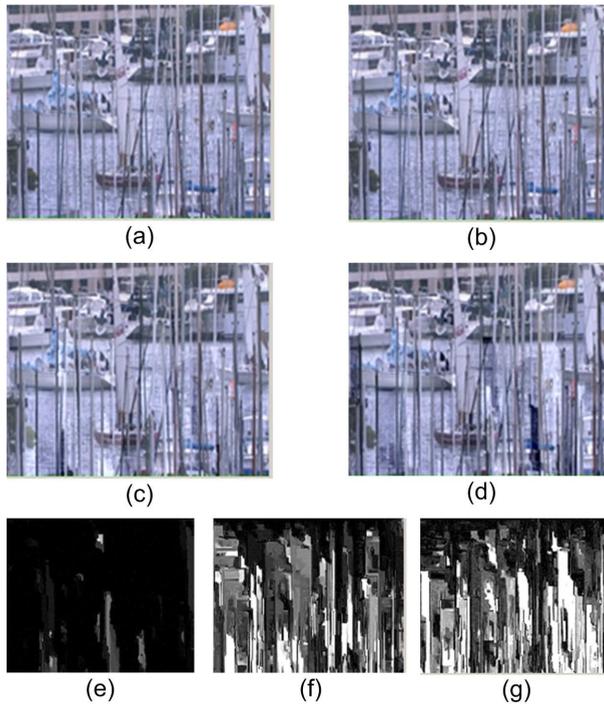


Figure 4.4 — Comparison of transform-domain reconstructed Harbor pictures of 176×144 pixels and corresponding error signal (factor 8 amplified). (a) Original picture. (b) Picture coded at $QP=6$. (c) Picture coded at $QP=18$. (d) Picture coded at $QP=28$. (e) Error for $QP=6$. (f) Error for $QP=18$. (g) Error for $QP=28$.

D. H.264/MPEG4-AVC scalable intraframe decoder

H.264/MPEG4-AVC intraframe compression differs from the preceding video coding methods such as MPEG-1/2/4, due to the presence of spatial prediction prior to transform coding. Intraframe H.264/MPEG4-AVC decoding requires, next to calculating the block-based residual information, a block-based prediction signal, which is calculated using integer arithmetic, to reconstruct the final pixel values. This prediction signal can be derived in the frequency domain, enabling scalable decoding in the transform domain, as proposed by Chen [95]. Kim [93] has used this technique to derive thumbnail-sized images from H.264/MPEG4-AVC intraframe-coded pictures. Although the results in [93] work for H.264/MPEG4-AVC up to main profile, the solution has two shortcomings. The first disadvantage is pixel-value drift as shown in Fig. 4.4, resulting from incorrect calculation of reference pixels employed for the calcu-



Figure 4.5 — *Drift effect on picture quality for a PiP picture with size 480×270 pixels due to non-exact predictor calculation and the associated propagation in the reconstructed signal based on that predictor.*

lation of the block-based spatial predictor. For thumbnail-sized pictures, this distortion may sometimes be perceptually acceptable, but for a PiP window of size 480×270 pixels, this distortion is perceived as annoying and therefore needs to be avoided, see Fig. 4.5. The second disadvantage of the frequency-domain decoding method of [95] is that it does not support the decoding of an 8×8 block size, which is deployed for profiles covered by the Fidelity Range Extension (FRExt) and utilizes low-pass filtering of the reference pixels prior to constructing the block-based prediction. From experiments using scalable decoding techniques as described in [95][93], it becomes clear that a computationally reduced H.264/MPEG4-AVC intraframe decoder must preserve the reference pixels, such that the decoder output is not contaminated by pixel drift. Furthermore, the method of Chen cannot be applied due to the above reasons.

Let us now summarize our requirements for audiovisual video navigation. The quality of the audiovisual navigation method relies heavily on the consecutive duration of the provided audio signal, whereas the derivation of the fast-search video navigation signal must be free from pixel drift. Further system and quality aspects and requirements of the navigation method are listed below.

1. *Normal-play fragment duration*: The normal-play duration is mainly determined by the perception of the audio signal, which differs for various content (music, speech, etc.).

2. *Scene-change detection*: Scene-change information is beneficial in order to adjust the start of a normal-play fragment, thereby optimizing the selection of sufficiently long meaningful audio fragments within the related time interval.
3. *PiP video decoding*: The PiP-based fast-search video navigation signal can be obtained by means of hardware or software decoding. The former is possible for high-end consumer platforms, which are equipped with dual-channel video decoding, while the latter solution is typically suited for low-cost consumer platforms. A PiP-based video navigation signal, which has a reduced spatial resolution compared to the original video, benefits from scalable video decoding, which is more cost-effective regarding the system resources, such as cycle consumption, bandwidth and memory footprint.
4. *Trick-play refresh-rate*: The trick-play refresh-rate is typically equal to the rendering rate. For trick play with high speed-up factors, it is advantageous to reduce the refresh-rate to allow an improved interpretation by the viewer. A reduced refresh-rate also lowers the computational load and decreases bandwidth, which is beneficial when decoding the fast-search video signal on a control processor.
5. *Video fragment processing*: The normal-play video fragments are constructed using multiple Group Of Pictures (GOPs). At the normal-play fragment boundaries, decoding artifacts may be visible depending on the absence of reference pictures. To avoid visible artifacts, the first and last GOP constructing a normal-play fragment (typically open GOPs) require modification (closing the open GOP structure) for avoiding decoding artifacts.
6. *Audio fragment processing*: Depending on the audio compression standard, padding samples may occur in an audio frame. Concatenation of non-sequential audio frames having padding samples causes an audio buffer violation, which can be prevented by applying proper signal processing in the compressed audio domain.

Based on the above requirements and system aspects, the next section presents a conceptual solution for the desired audio-enhanced dual-window based navigation method.

4.3 Concept of audio-enhanced dual-window video navigation

This section provides an outline of the audio-enhanced dual-window video navigation signal. First, we discuss the involved temporal subsampling for de-

iving audiovisual fragments from the normal-play signal and the associated fast-search video signal. Second, two solutions are presented for the involved signal processing.

4.3.1 Temporal subsampling of dual-stream video signal

The concept of audio-enhanced dual-window navigation is based on detailed normal-play audiovisual information, combined with fast-search video navigation information. The derivation of both visual navigation signals is based on temporal subsampling of the normal-play video sequence, which is depicted in Fig 4.6(a). The first signal, depicted in Fig. 4.6(b), constructs the primary video window on the basis of normal-play fragments, featuring also the corresponding audio information. The second information signal constructs the PiP-based fast-search video navigation window, of which the time sampling is depicted in Fig. 4.6(c). Hereby, $t_{s,k}$ indicates the time location relative to time $t_{n,0}$, which denotes the start point of navigation of the normal-play signal. Time sampling refers to a temporal picture selection process, in which the normal-play pictures

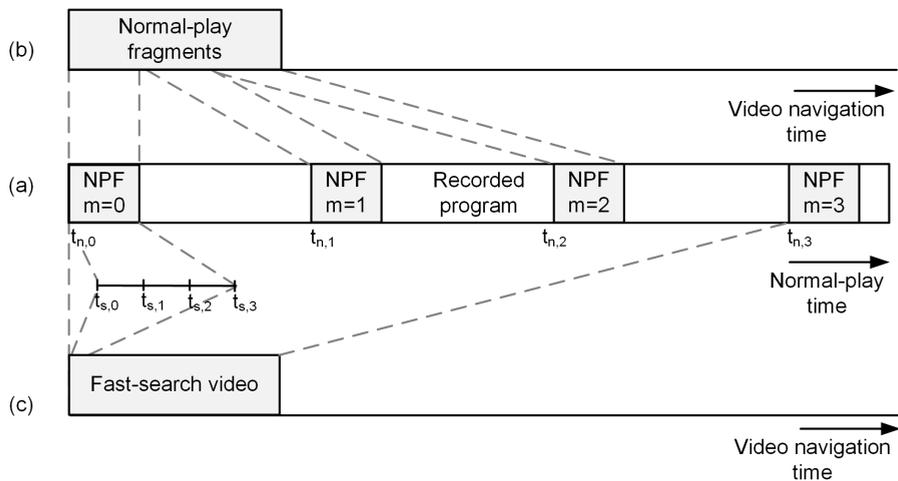


Figure 4.6 — Time line for extracting dual-window video navigation (indexes are explained in the text). (a) Recorded MPEG-compressed program with indexed normal-play fragments (NPF). (b) Selected normal-play fragments which are appended as a sequence. (c) Derived fast-search navigation signal using intra-coded pictures.

are selected that coincide with the time moments for fast-search video navigation, according to the relation

$$t_{s,k} = \frac{kP_s}{f}. \quad (4.1)$$

Parameter P_s is the relative playback speed for the fast-search video navigation signal, $k = \{0, 1, 2, \dots, K\}$, an index revealing the re-used normal-play pictures constructing the trick-play signal and f being the television frame rate. The corresponding normal-play fragments have a normal-play fragment duration time T_{np} , where $t_{n,m}$ denotes the m -th normal-play fragment start position relative to $t_{n,0}$. Hereby, parameter $m = \{0, 1, 2, \dots, M\}$, represents an index pointing to the re-used normal-play fragments, which is specified by

$$t_{n,m} = mT_{np}P_s. \quad (4.2)$$

When considering a 25-Hz television frame rate, a normal-play GOP length of 12 and the relative playback speed $P_s = 12$, which is motivated based on a practical value for an intraframe refresh-rate of 2 Hz. In this way, typical successive values for the trick-play fast-search signal $t_{s,k}$ become according to Equation (4.1) integer multiples of 0.48 seconds, while typical successive values for the normal-play fragments $t_{n,m}$ become according to Equation (4.2) integer multiples of 36 seconds when $T_{np} = 3$ seconds.

4.3.2 Conceptual solutions for audio-enhanced dual-window video navigation

Similar as with the video navigation solutions, discussed in Chapter 3, there are two options for efficiently deriving the involved video and audio signals. First, signal derivation during video navigation playback only and second, signal derivation divided over record and navigation playback mode. Both options are briefly analyzed with respect to their complexity, so that the best concept can be chosen.

A. First concept: signal derivation during video navigation playback

In the first approach, the navigation signals are derived during playback only. In such an approach, we use MPEG decoding for the normal-play fragments and scalable MPEG decoding for deriving the trick-play video signal, see Fig. 4.7(a). In this figure, at the left side, two MPEG-compressed data fragments, i.e. one I-picture and one GOP, are retrieved from the storage medium and each fragment passes through an MPEG-2 demux to access pictures and audio from the resulting video/audio elementary streams. The first fragment contains an intra-coded picture for deriving the fast-search navigation signal,

while the second fragment contains the audiovisual information for detailed rendering. The following signal processing steps are required for further processing of the selected fragments.

- *Video pre-processing*: This step involves removal of predictive-coded pictures, referring to non-existing reference pictures, thereby avoiding decoding artifacts due to an open GOP structure.
- *Audio pre-processing*: This involves stream adaptation resulting from making a series of concatenated audio fragments to create a new audio stream that should be MPEG compliant for playback.
- *Fast-search video signal processing*: For deriving PiP-sized pictures, the fast-search video elementary stream, which is based on only I-pictures, is scalable MPEG decoded for deriving pictures of smaller size.

For the normal-play fragments, the video elementary and audio elementary streams are fully decoded, resulting in decompressed normal-play pictures and decompressed audio samples, respectively. Finally, the scaled fast-search decoded I-pictures are mixed with the decoded normal-play fragment pictures, resulting in the final dual-window video navigation screen.

Complexity. Let us derive a performance estimation and thus an impression of the computational load on the key system resources. The outcome of this estimation is summarized in Table 4.1. The navigation operates at a 25-Hz video frame rate for the European situation. The figures in Table 4.1 apply to a situation that the dual-window video screen is derived from a 720×576 picture size (SD) and 4:2:0 sampling format for the video signal and involves Main Profile MPEG-2 decoding. The PiP is derived in a scalable manner from normal-sized I-pictures, resulting in a PiP-size of 180×144 pixels and 4:2:0 sampling format. We have omitted the audio signal, as the video signal is the dominant factor in complexity.

Using previous figures, dual-window video navigation is constructed from two independent images with the previous resolutions, sampling format and frame rate. This results in an uncompressed video bit rate of 124.42 Mbit/s and involves a memory capacity requirement of 608 kBytes per full-color main image frame ($720 \times 576 \times 1.5$ pixels), while PiP has a bit rate of 7.56 Mbit/s and involves a memory capacity requirement of 38 kBytes per full-color frame ($180 \times 144 \times 1.5$ pixels). The total bit rate for such dual-window video screen at 25 Hz (full frame rate) equals 441 Mbit/s and involves a memory capacity of 2,310 kBytes.

The concept is based on playback of normal-play fragments of sufficient duration to also include audio. The audio signal is decoded in the same way the normal playback of a video program. The additional processing related to the

fast-search PiP signal is only a small fraction of the normal-play processing complexity, as seen from the above figures. The decoding complexity of the PiP signal is expected to be modest, of which implementation details will be discussed later.

Table 4.1 — Performance estimation on system resource utilization for audio-enhanced dual-window navigation with full frame-rate PiP signal derived from SD video.

Concept	System resources			
	Decoder bandwidth	Mixer bandwidth	Memory Capacity	DSP/CPU cycle load
Playback proc. only	441 Mb/s	257 Mb/s	2,310 kB	Modest
Record & playback proc.	411 Mb/s	257 Mb/s	2,310 kB	Low

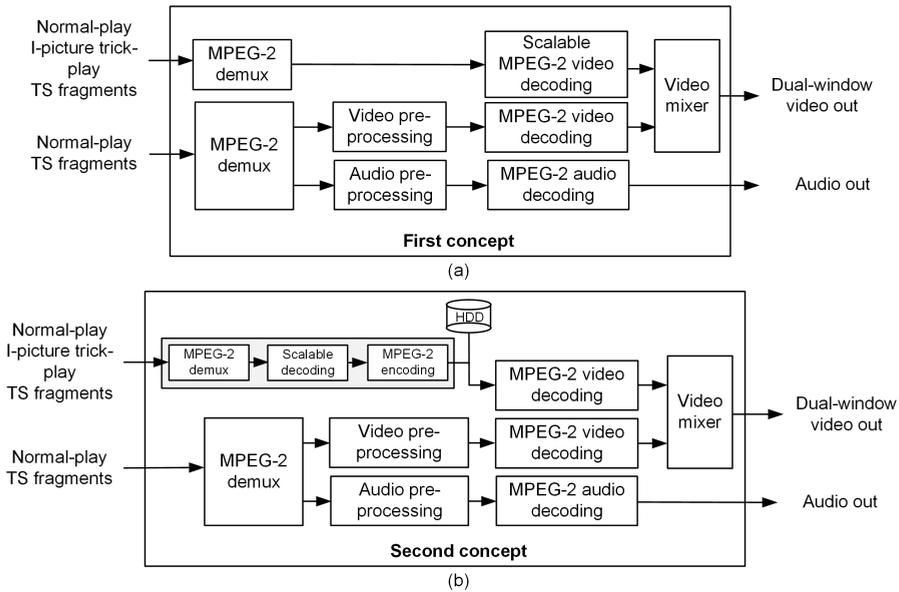


Figure 4.7 — Conceptual signal processing for audio-enhanced dual-window video navigation. (a) All navigation signals are decoded during navigation playback. (b) All navigation signals are decoded during navigation playback, but the fast-search signal is downsampled during record and stored in MPEG-2 format.

B. Second concept: video signal processing divided over record and navigation playback

In the second approach, the fast-search PiP images are already derived during recording and stored in MPEG format. This approach is visualized in Fig. 4.7(b). The normal-play fragments are processed with the steps as indicated in the previous concept. We now indicate only the difference for deriving the fast-search signal, which is an extra pre-processing block at recording stage in the diagram of Fig. 4.7(b).

- *Fast-search PiP processing:* During recording, the intraframe-coded pictures are extracted, scalable MPEG decoded and then re-encoded as a separate MPEG intra-coded picture stream, stored on the recording medium.

During navigation, the fast-search video information signal is MPEG-2 decoded and mixed with the decoded normal-play fragment pictures, resulting in a final dual-window navigation screen.

Complexity. The performance estimation and key system resource usage is to a large extent similar to the first concept. Let us now focus on the difference. The PiP video signal is derived during record from intraframe-compressed pictures on the basis of scalable MPEG-2 decoding resulting in a resolution of 180×144 pixels and 4:2:0 color sampling format. During record, these images are MPEG-2 compressed at a bit rate of 15 Mb/s, and stored on the medium. This bit rate is rather high, because only intra-coded pictures are processed. The total bandwidth for such dual-window video screen at 25 Hz (full frame rate) equals 411 Mbit/s and involves a memory capacity of 2,310 kBytes. There is only a small bandwidth difference of 30 Mb/s compared to the first concept, which is the difference bandwidth between the required 45 Mb/s for the PiP signal in the first concept and 15 Mb/s that is required for the PiP in the second concept.

Concept conclusion: When comparing the two proposed concepts with respect to their system requirements, they have almost equal system load in terms of bandwidth and memory capacity. However, in the second concept, the fast-search signal is a 16 times smaller in picture size than in the first concept. Moreover, the compressed fast-search bit stream in the second concept is also 16 times lower than in the first concept. The execution complexity of the fast-search decoding processing software will be downscaled with a similar factor. Furthermore, the intra-coded pictures used for deriving the fast-search navigation signal have a frame rate depending on the GOP length, which equals typically 2 Hz, thereby enabling slow-speed processing for each picture. These arguments make the second navigation concept based on pre-scaled fast-search

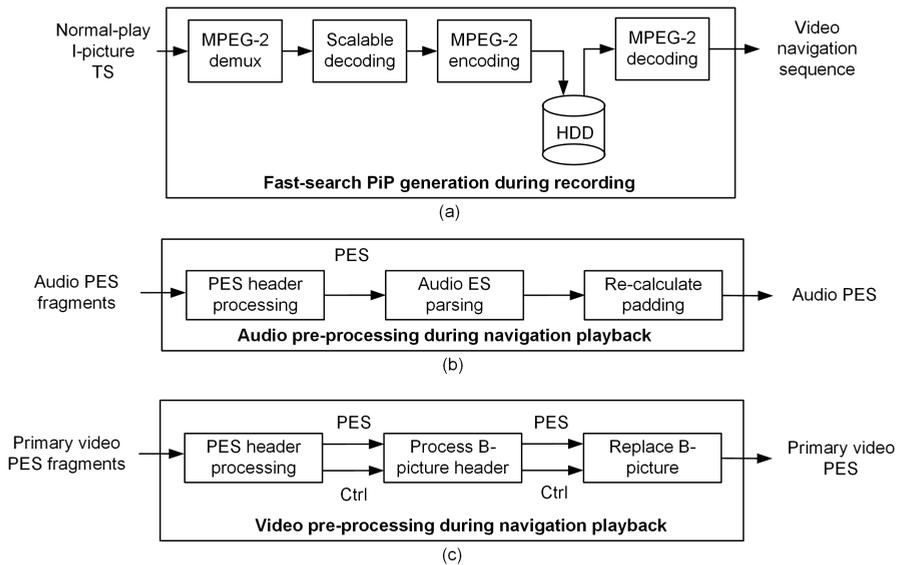


Figure 4.8 — Essential signal processing steps for deriving multi-signal navigation information. (a) Processing steps for deriving a PiP-sized video navigation signal. (b) Processing steps for normal-play audio fragment processing. (c) Processing steps for normal-play video fragment processing.

images the preferred solution. This choice is further motivated when considering more complex compression standards such as H.264/MPEG4-AVC.

4.4 System integration and implementation

This section aims at implementing the adopted the second navigation concept of the previous section, which was chosen after the analysis of the concepts. First, the section starts with further detailing the involved audio and video processing required for the adopted concept. Second, the involved signal processing is mapped onto a functional PVR block diagram, re-using or extending the functional blocks which have been already detailed in Chapter 3.

4.4.1 AV algorithm implementation of chosen navigation concept

For the second concept, Fig. 4.8 depicts the main involved signal processing steps for the three individual navigation information signals. Fig. 4.8(a) depicts the main processing steps for deriving the PiP-sized fast-search video

navigation signal, which is conducted during recording. Starting at the left, the normal-play video sequence is demultiplexed providing access to the intra-coded pictures constructing the normal-play video sequence. This video elementary stream is parsed and partially scalable decoded. These decoded normal-play pictures are MPEG-2 intraframe (re-)coded and are locally stored. At the right, during navigation playback, these PiP-sized MPEG-2 intra-coded pictures are decoded, forming the fast-search navigation information signal.

Fig. 4.8(b) depicts the main pre-processing steps for processing the MPEG-compressed audio information during video navigation. At the left, the demultiplexed Packetized Elementary Stream (PES) access units are pre-processed, involving time-stamp adaptation to the local navigation time base and potential adaptation of available padding slots, to comply to the MPEG data format. For this adaptation, the audio PES is parsed, scanning for audio start codes and audio header inspection. For the situation that padding is required, already available padding slots are removed and new padding slots are calculated. In this way, an MPEG-compliant audio sequence is derived in the MPEG-2 compressed domain.

Fig. 4.8(c) depicts the main pre-processing steps for processing the MPEG-compressed video information during video navigation. At the left, the demultiplexed PES access units are pre-processed, for time-base adaptation and removal of predictive-coded pictures, that cannot be fully decoded due to the absence of the associated reference pictures. This processing involves discontinuity detection in the original Decoding Time Stamps (DTS) and Presentation Time Stamps (PST), which are present in the PES header. For the situation that a discontinuity occurs at the boundary of two concatenated normal-play GOPs, the potentially available B-coded pictures are replaced by repetition pictures, i.e. pictures repeating the last decoded reference picture. In this way, decoding artifacts are avoided, which would deteriorate the video navigation quality.

Let us now detail the processing algorithms. Figure 4.9(a) and (b) visualize the algorithmic video and audio processing steps, respectively. The fast-search video signal processing involved in scalable MPEG decoding was already discussed in Section 3.6.2 and is omitted here.

Video processing: The normal-play fragment-based video navigation signal is constructed on the basis of concatenating normal-play GOP-based fragments. As these GOPs typically contain bi-directional predictive-coded pictures (B-type) potentially referring to past and near future, these decoded pictures regularly contain errors after decoding due to the absence of reference pictures. The objective of this video processing is the removal of these B-pictures and replace them by a repetition picture, repeating the last decoded reference picture. In this way, severe video decoding artifacts are avoided. The discontinuity detection requires access to the PES header, which contains the normal-play DTS

and PTS time-stamp values. The time difference between these time stamps are multiples of the reciprocal frame rate, which is typically 2 or 3 frame periods. Concatenation of GOPs, that have a temporal distance of seconds, leads to a jump in the temporal distance which is far beyond the 2–3 frame period interval (`test discon true ?`) in Fig. 4.9(a)). For such a situation, the B-pictures are replaced with B-repetition pictures equipped with a *temporal reference*, which is derived from the removed B-pictures. Furthermore, the DTS and PTS values are updated corresponding to the navigation-playback time base, thereby enabling smooth playback.

Audio processing: Similar to the video navigation, audio navigation is constructed by concatenating frames of audio intervals. Such an audio frame may be equipped with a padding slot, which depends on the applied sampling rate, see Section 2.2.3. Concatenation of audio fragments corresponding with normal-play GOPs may result in a compliancy violation, due to the absence or frequent occurrence of these padding slots. This problem is solved by the algorithmic steps in Fig. 4.9(b), involving first the detection of padding slots (`pad. req. == True ?`), followed by padding slot averaging over the received frames. For the situation that the calculated number of average slots does not correspond to an integer value (`pad. slot == True ?`), a padding slot is inserted. This calculation requires a continuous adaptation. Finally, the corresponding DTS value is modified according to the navigation playback time (Set DTS/PTS to `trick-play time base`), while preserving the audio time synchronization to the video signal.

4.4.2 Functional block diagram of the chosen concept

This section briefly discusses the architecture of the implementation of the chosen concept for video navigation. We aim at a functional block diagram which is comparable with the diagrams presented in Chapter 3. It will become clear that there are significant commonalities between the diagrams of the navigation concepts of the previous chapter and the adopted concept in this chapter. Figure 4.10 depicts the PVR functional block diagram, extended with the additional functionality. The signal paths for the various signals in Fig. 4.10 required during navigation are indicated in the figure caption. When the switch at the bottom of Fig. 4.10 is set to path (c), this resembles navigation playback proposed as in Chapter 3, while when the signal path (c) in combination with signal (d) is used, this resembles audio-enhanced dual-window based video navigation. The signal paths (a) and (b) correspond to normal viewing condition and time-shift recording, respectively.

Careful inspection of this block diagram reveals that most of the functional

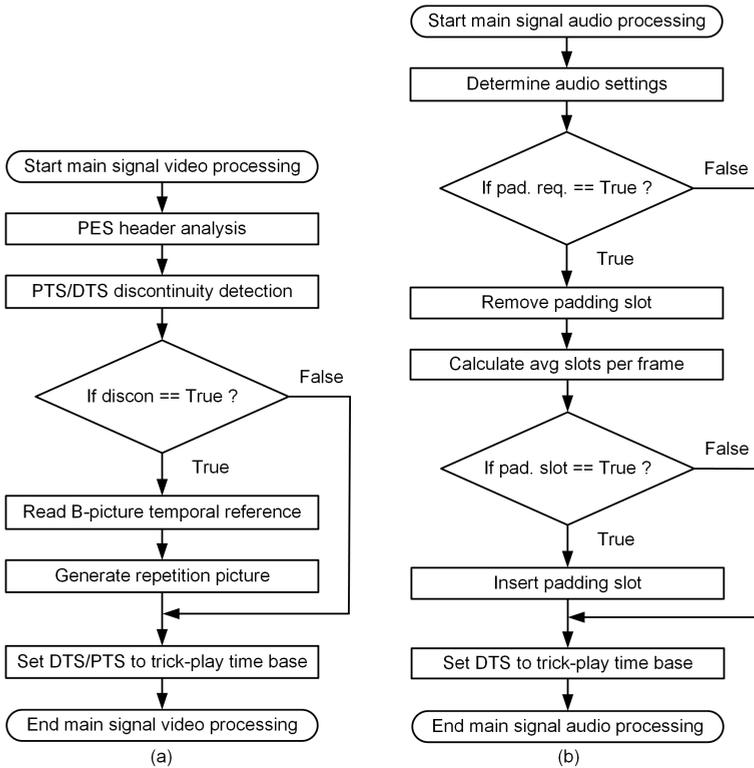


Figure 4.9 — Video and audio processing steps for navigation. (a) Flowchart of the video algorithm. (b) Flowchart of the audio algorithm.

blocks are identical to the blocks from the diagram of Section 3.7. In contrast, the blocks surrounded by a dashed line contain novel or modified processing and extend the architecture for dual-window navigation. Let us briefly discuss the common blocks and the two dashed blocks.

- *Common blocks:* During Transport Stream (TS) recording, Characteristic Point Information (CPI) is derived, which amongst others allows tracking of the intraframe-coded pictures of the recorded program. This CPI block is functionally identical, but somewhat different from the solution in the previous chapter. The difference is that the spatially reduced PiP is also intra-coded, but with a frame-based constraint instead of a slice-based constraint. This information and related parameters are stored in the metadata database in a similar way as in the previous chapter. The audiovisual information stream is written or retrieved from the storage

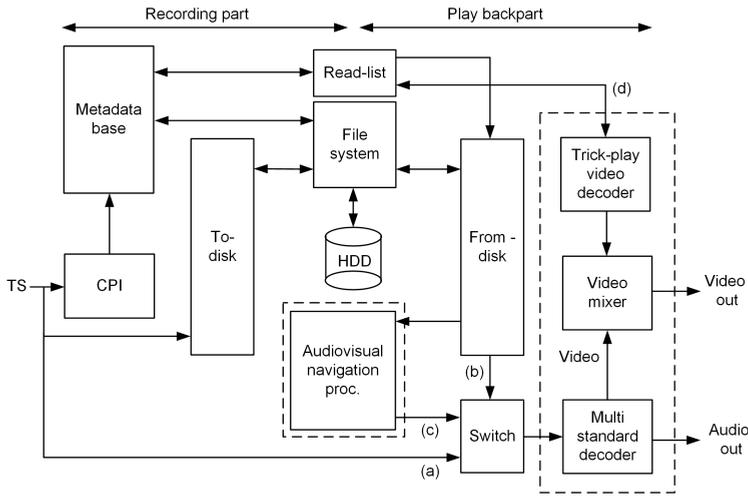


Figure 4.10 — Functional block diagram for implementing audio-enhanced dual-window video navigation. Signal path (a) is for real-time viewing. Signal path (b) is for time-shift recording. Signal path (c) is for conventional trick-play or contains the main video signal in case of double-window based trick play. Signal path (d) is for the PiP in case of double-window fast-search trick play.

medium via the to-disk block and from-disk block. For the various playback situations, the read-list block determines, via the metadata database, the medium access points of the individual information streams and controls the from-disk block. This block retrieves the normal-play TS, or in case of trick play, the normal-play TS fragments for trick play. The TS fragments for trick play contain a multiplex of audiovisual information, which are required for the audiovisual signal playback for the main window.

- *Dashed block: audiovisual navigation proc.:* This modified navigation processing block in Fig. 4.10 is not only dedicated to video, but also contains audio processing. It performs an MPEG-2 demultiplexing operation to access the individually compressed video and audio access units. The video processing operating in the compressed domain, replaces pictures with repetition pictures, as explained above. The selection of the compressed normal-play audio frames are associated with the video fragments, thereby enabling audio-based navigation in parallel with video navigation for the main window. The processing also involves the padding insertion as previously discussed.

- *Dashed block: Trick-play decoding and mixing:* This second dashed block performs the involved video decoding and associated video mixing of the fast-search signal and the main signal. The pictures constructing the fast-search trick-play signal are retrieved from the metadata database.

4.5 Computational reduced H.264/MPEG4-AVC intraframe decoding

Up to this point, the work on video navigation techniques has been based on MPEG-2 compressed signals. During this research, HD video communication and storage has become popular and has introduced follow-up standards for High-Definition (HD) compressed video. For HD video compression, the H.264/MPEG4-AVC standard has been widely accepted and applied. This standard is also based on motion-compensated DCT coding and achieves approximately a factor of two higher compression compared to MPEG-2 for a similar quality. This growth in video compression is achieved at the expense of approximately four times higher complexity. For the details of the standard, we refer to Chapter 2, in particular Section 2.3.

The question arises whether the proposed navigation techniques in this thesis are also applicable to this new standard. This section elaborates on this aspect and will show that it is indeed possible to employ the proposed navigation techniques also to H.264/MPEG4-AVC-compressed video sequences. This positive conclusion is expected considering that the standard is also based on DCT coding in combination with motion-compensated predictive coding. Also in H.264/MPEG4-AVC, GOPs of consecutive video frames are constructed, starting with an intra-coded picture. However, the concept of predictive coding of successive images is different and is more intensively used, so that it is more difficult to collect reference frames. This all together makes video navigation clearly more complicated for implementation. This motivates the discussion in this section on complexity-reduced decoding of H.264/MPEG4-AVC intra-coded pictures.

We have discussed scalable MPEG-2 decoding as an option for complexity reduction to facilitate the implementation of embedded navigation. For H.264/MPEG4-AVC, such techniques have also been explored and this was briefly addressed in Section 4.2. It was concluded that this solution suffers from pixel drift, leading to a clear deterioration of the image quality, which is not acceptable for PiP images of reasonable size. In this section, we therefore present two alternative concepts for complexity-reduced H.264/MPEG4-AVC intraframe decoding that avoid pixel drift, enabling the derivation of PiP-sized images with sufficient quality for fast-search navigation. The first concept exploits partial

calculation of the spatial prediction block, thereby ensuring exact calculation of the pixels employed for spatial prediction. In this way, drift can be circumvented. The pixel locations that are not exactly calculated, obtain a pixel value based on duplicating the nearest exact reference pixel. This leads to the second concept, which is an add-on to the first concept. Whereas in the first concept the non-reference pixels are also transformed and decoded as a residual signal, the second concept omits this transform calculation for the non-reference pixels to further reduce the complexity. This approach is acceptable as the target fast-search resolution for navigation is clearly lower than the original signal resolution.

4.5.1 Partial reconstruction of the prediction block

H.264/MPEG4-AVC intraframe compression employs a block-based prediction signal to reduce a set of pixel values prior to DCT transformation. For the construction of this block-based prediction signal, a set of reference pixels is employed, which are located at the bottom rows and at the outer-right columns of the neighboring 4×4 , 8×8 or 16×16 pixel blocks. Since H.264/MPEG4-AVC intraframe decoding (see Fig. 4.2) is similar to MPEG-2 and a reduced spatial resolution is targeted, it is plausible that during decoding, the complexity of computing the transform block and spatial prediction block can be reduced by partial calculation (like in scalable MPEG-2 decoding). In this approach, pixels not involved by the spatial prediction and not required for the derivation of the lower resolution picture, are omitted from calculation.

A. 1st concept: Partial calculation of block-based spatial predictor

Up to main profile, H.264/MPEG4-AVC intraframe compression deploys spatial prediction on the basis of a 4×4 or 16×16 pixel block size. For profiles based on Fidelity Range Extensions (FRExt), an additional 8×8 block size is employed. Similar as to the 4×4 block size, with the 8×8 block size a spatial prediction is conducted prior to the 8×8 transformation. Unlike the calculation of a 4×4 predictor, the pixels involved for calculating an 8×8 predictor are low-pass filtered prior to the calculation of the block-based predictor. Both 4×4 and 8×8 block sizes use 9 spatial prediction techniques, while the 16×16 block size uses only 4 spatial prediction techniques. Depending on the spatial prediction mode, arithmetic operations are employed to calculate the final block-based prediction signal, using neighboring reference pixels. Furthermore, potential spatial prediction pixels, located at the bottom row and outer-right column of a 4×4 or 8×8 pixel block, may also be used inside such a predictor block, see the gray color locations in Fig. 4.11. Note that the index at the various pixel locations in Fig. 4.11 corresponds to the pixel index, as deployed in the H.264/MPEG4-AVC reference software model JM18.2 to calculate the predic-

tor, where the three most right blocks have the origin at the upper-left corner, while the other blocks have the origin elsewhere. The caption indicates the prediction mode. For spatial prediction, with an exactly calculated pixel we mean a pixel that is calculated with all required calculations as specified by the standard. We call such pixels also reference pixels. The five shown prediction modes of Fig. 4.11 have clearly less exact reference pixels than the amount of pixels in the block, so that the duplication provides a direct computation reduction. However, there are up to 9 prediction modes of which 4 modes are implemented without computation reduction. From these 4 modes, the vertical and horizontal prediction modes duplicate reference pixels only, involving no additional computations. From the remaining 2 modes, the DC-based prediction mode calculates the average of the available reference pixels, which cannot be reduced. The last remaining mode is the diagonal-down-right prediction mode, which employs the calculated reference pixel within the block for repetition already. As a conclusion, only the 5 modes from Fig. 4.11 benefit from complexity reduction. Figure 4.11 indicates the gray prediction pixels that are exactly calculated. All white pixels are obtained by repetition. Although the objective is to calculate only the prediction pixels at the bottom row and outer-right block boundaries, also a substantial amount of non-boundary pixels are still correctly calculated, which limits the degradation of the block-based predictor. Besides a 4×4 and 8×8 block-based predictor, H.264/MPEG4-AVC also employs 16×16 block-based prediction, see Fig. 4.12. From the four 16×16 prediction modes, only the plane prediction mode as shown in Fig. 4.12, benefits from the reduced calculation approach. Again the white pixel positions ob-

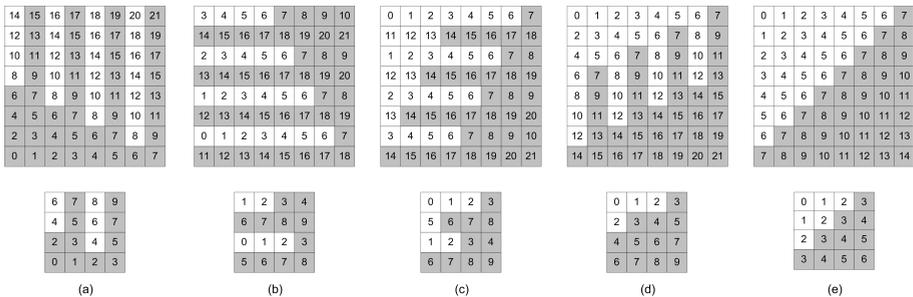


Figure 4.11 — Block-based predictor for 4×4 and 8×8 pixel prediction modes in H.264/MPEG4-AVC, utilizing partial calculation. Gray pixels are exactly calculated, whereas white pixels are duplicates of the nearest exact neighbor. (a) Horizontal-down prediction. (b) Vertical-right prediction. (c) Vertical-left prediction. (d) Horizontal-up prediction. (e) Diagonal-down-left prediction.

Table 4.2 — *Computational reduction in percentages for various intraframe 4×4 and 8×8 block-based predictor calculations.*

Prediction mode	Calculation reduction for 4×4 block size (%)			Calculation reduction for 8×8 block size (%)		
	Add	Mult	Div	Add	Mult	Div
Diagonal down left	45	42	42	46	46	46
Vertical right	27	16	30	30	21	31
Vertical left	36	20	40	42	27	45
Horizontal up	40	50	50	67	42	71
Horizontal down	24	16	30	30	21	31

tain a predictor value based on the nearest gray pixel location. It should be noted that the first concept is executed with the standard H.264/MPEG4-AVC de-blocking filter switched-off, to save on complexity and because the navigation picture resolution is limited. The obtained computational reduction for the 16×16 plane-based predictor equals 75% of the involved arithmetic operations: addition, multiplication, subtraction and division. Figure 4.12 reveals that not only the shown row and column predictor pixels are calculated exactly to avoid drift, but also 9 predictive samples located at a subsample grid of 4×4 pixels.

In this way, the PiP quality is optimized, when deriving a factor four horizontally and vertically downsampled grid. Table 4.2 shows the computation reduction in terms of percentages for the involved arithmetic operations for the 5 spatial prediction modes of Fig. 4.11.

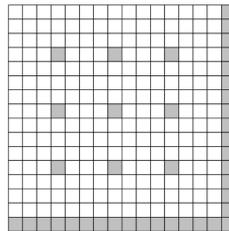
**Figure 4.12** — *Reconstructed 16×16 plane-based prediction in H.264/MPEG4-AVC for reduced calculation in intraframe decoding. Gray pixels are exactly calculated.*

Table 4.3 — *Number of operations (# ops) required for regular and partial calculation of a 4×4 and 8×8 block in the H.264/MPEG4-AVC inverse DCT transformation.*

Transform block size	Operation add, shift	# Ops / block normal dec.	# Ops / block partial calc. dec.	Reduction
4×4	+	64	49	24%
	\gg	16	13	50%
8×8	+	512	365	29%
	\gg	160	118	26%

B. 2nd concept: Partial removal of transformation of the residual block

This part refers to the add-on concept to further reduce the involved computations of the intraframe H.264/MPEG4-AVC decoding. When partially calculating the spatial prediction block, certain pixels cannot be properly reconstructed. The complexity can be further reduced, by employing a partial inverse transformation instead of a full transformation, thereby restricting the IDCT computation to only those pixels that are located at the block bottom-row and outer-right column, which are always available, see e.g. Fig. 4.11. For the 4×4 and 8×8 IDCT, this principle of calculating only the bottom-row and outer-right column pixels is always applied. This yields a reduction in the amount of operations, as shown in Table 4.3. The 2D IDCT deployed in H.264/MPEG4-AVC can be written as a matrix calculation $X = A^T Y A$, where matrix Y contains the transform coefficients, A denotes the IDCT transform matrix and X contains the output result of the inverse 2D transform. From the notation $X = A^T Y A$, it is evident that the 2D IDCT is calculated in two separable stages. A computational reduction is obtained only in the second stage, when the transform is only calculated for matrix elements located at potential spatial prediction locations, i.e. bottom-row and outer-right column. This partial inverse DCT transformation is used for decoding the residual signal that is added to the prediction signal.

4.6 Experimental results

In this section, we discuss four aspects of the proposed navigation system. First, scalable MPEG-2 decoding performance is presented, obtained for the algorithmic simplification proposed in Section 3.6.2, using IDCT computation reduction via adds and shifts. Second, for both H.264/MPEG4-AVC intraframe decoding concepts, the final picture quality is presented. Third, for these con-

cepts, the complexity in terms of execution cycle load performances is given. Fourth, the proposed audio-enhanced dual-window navigation is perceptually evaluated by a small test panel.

The experiments are performed with the following experimental setup. For scalable MPEG-2 decoding, the algorithmic simplification is incorporated in the MPEG-2 video software reference model (ISO/IEC 13818-5), to evaluate the final intraframe-decoded picture quality.

The reductions for calculating the partial spatial predictor block and the partial IDCT computation, as discussed in Section 4.3, are incorporated in the H.264/MPEG4-AVC reference software model JM18.2, in order to evaluate the final picture quality. This modified test software model is executed on a regular PC platform. In the first concept, the intraframe decoder employs a partial calculation of the spatial prediction block and omits the de-blocking filter. In the second concept, the first concept is extended by further reducing the decoding complexity by the partial calculation of the residual signal.

4.6.1 Picture quality of scalable MPEG-2 SD video decoding

This subsection presents the subjective and objective picture quality of the fast-search navigation based on scalable MPEG-2 decoding. Figure 4.13 shows a set of 8 QCIF-resolution pictures obtained with scalable MPEG-2 decoding. The subjective quality is good, especially when considering that these playback images are shown in fast-search mode, which makes detailed scene analysis by the viewer impossible, as they only serve the purpose of obtaining a global overview of the video contents by the viewer. Table 4.4 shows the objective picture quality measured with scalable MPEG-2 decoding containing our algorithmic simplification. The label of each image corresponds to the picture labels in Fig. 4.13. The PSNR varies between 26–32 dB, and depends considerably on the image content. Although a PSNR of 26 dB is at the lower bound for normal-play video content, for fast-search video information this is certainly acceptable for the employed navigation playback speeds (each picture is only shown for 240 ms with $P_s = 12$).

4.6.2 Picture quality for H.264/MPEG4-AVC decoding concept 1 & 2

For modern storage devices and HD broadcasting TV, the navigation will be based on the H.264/MPEG4-AVC standard. In the previous section, two concepts were proposed aiming at a partial calculation of the spatial prediction block and partial IDCT calculation. Figure 4.11 and Fig. 4.12 in Section 4.3 visu-

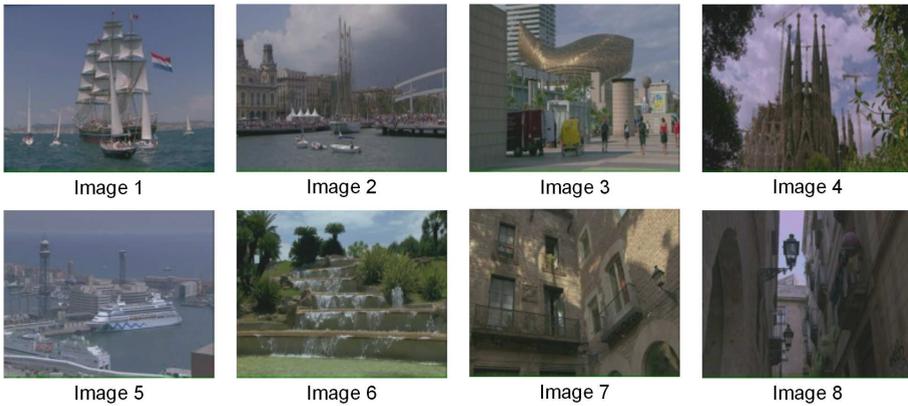


Figure 4.13 — Visual results for QCIF-resolution PiP images obtained by scalable MPEG-2 decoding of SD intra-coded pictures, corresponding to fast-search playback speed with $P_s = 12$.

alize in gray color the exactly calculated pixels according to the standard and in white color the replicated pixels. We first discuss the quality of these concepts and later the execution complexity.

Concept 1

Table 4.5 indicates the objective picture quality for full-HD decoded pictures, based on Concept 1 (partial spatial prediction without de-blocking filter). The quality of this navigation signal is first evaluated at full-HD resolution, to mimic the case that the PVR platform is equipped with a video scaler. Examples of visual results obtained by Concept 1 are depicted in Fig. 4.14 and Fig. 4.15. The shown error-difference signal is amplified by a factor 64 for visualization only, as the average distortion is below unity. The printed pictures from Fig. 4.14 and Fig. 4.15 have also been visually inspected on a full-HD receiver

Table 4.4 — Objective image quality for QCIF-resolution PiP images obtained by scalable MPEG-2 decoding of SD intra-coded pictures, corresponding to fast-search playback speed with $P_s = 12$ (the image index corresponds to Fig. 4.13).

Image	1	2	3	4	5	6	7	8
PSNR (dB)	32.24	28.99	27.08	27.92	28.68	26.91	28.65	28.25

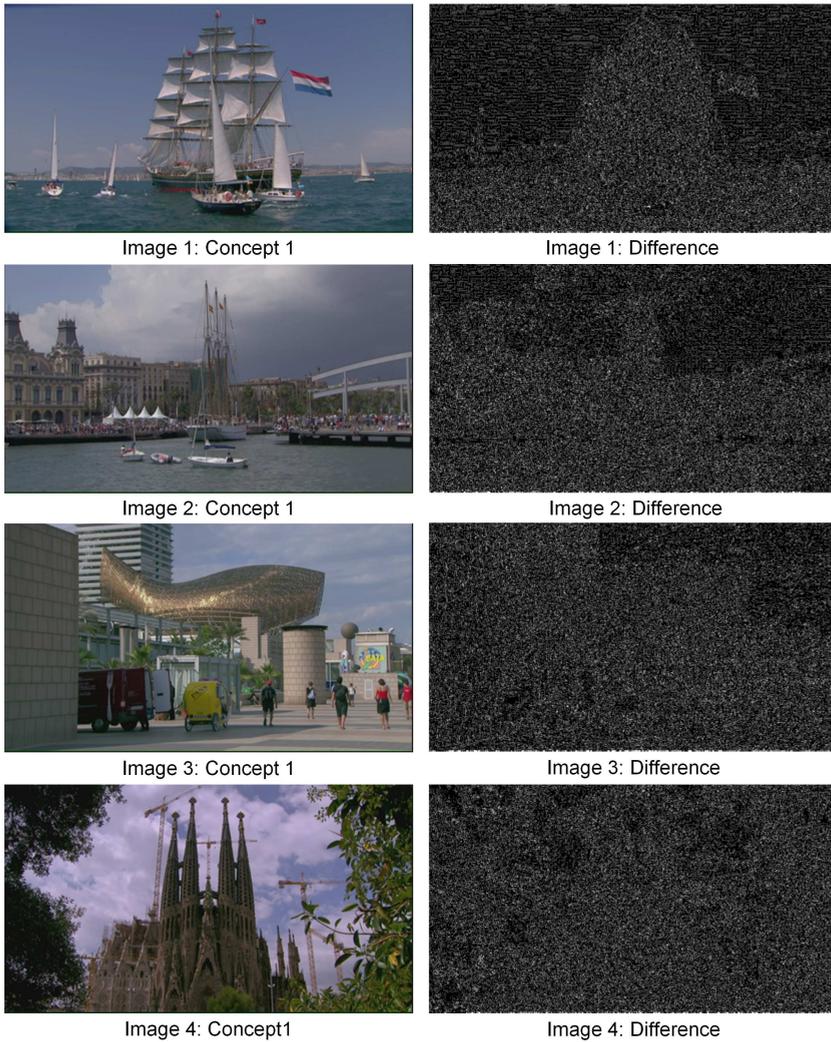


Figure 4.14 — *Final PiP image quality obtained from decimation-filtered and down-sampled intra-coded full-HD images according to Concept 1. The left column shows the derived PiP images. The right column shows the difference between the left image and a PiP image obtained with full-HD standard decoding followed by decimation-filtered downsampling. For visualization, the error is a factor 64 amplified.*

via HDMI reception. The navigation pictures were viewed at a PiP resolution of 480×270 pixels. The qualities of these pictures appear to be quite good and well suited for navigation purposes. The printed pictures already indicate that there are no severe distortions. The right column in Fig. 4.14 and Fig. 4.15 shows that the coding error is equally distributed over the available textured areas in the picture and that there are no extreme image degradations. Although the intraframe-decoded picture quality is negatively influenced by the pixel replication process, the objective picture quality drops considerably but not dramatically, staying in the 30-dB range (31–37 dB), see Table 4.5, as compared to the quality range without pixel replication, which is indicated at the right of the discussed column (37–40 dB). It can be seen that the absolute quality of the decoding process of H.264/MPEG4-AVC is dominant in the reference quality and the complexity reduction causes only a minor image quality degradation. This is confirmed by the small average error indicated at the left column of Table 4.5, despite the spurious maximum errors in the signal. Note that due to the absence of a de-blocking filter, also small deviations in the black regions occur.

However, a good quality PiP can be obtained, see most-right column of Table 4.5 for fast-search. The PiP image was obtained with a partially decoded image and applying a 7-tap horizontal and 5-tap vertical decimation filter. The evaluated quality in the range of 39–46 dB is derived by comparing a decimation-filtered downscaled full-HD decoded picture, with the PiP

Table 4.5 — *Objective quality of PiP pictures derived from decimation-filtered full intraframe-decoded H.264/MPEG4-AVC pictures involving partial spatial prediction block calculation, replication of calculated prediction pixels and omitted de-blocking filter.*

Test image (QP=28)	Partially dec. trick play Luminance			Full dec. Luminance	PiP Luminance
	Average error	Maximum error	PSNR (dB)	PSNR (dB)	PSNR (dB)
Image 1	1.87	169	36.81	40.01	46.41
Image 2	2.10	189	36.25	39.42	45.49
Image 3	2.48	213	33.90	38.85	43.20
Image 4	3.04	198	31.45	38.17	39.85
Image 5	1.84	169	37.99	40.08	48.83
Image 6	3.20	189	32.15	37.62	40.05
Image 7	3.17	142	33.32	37.64	41.02
Image 8	2.03	178	37.70	40.38	46.33

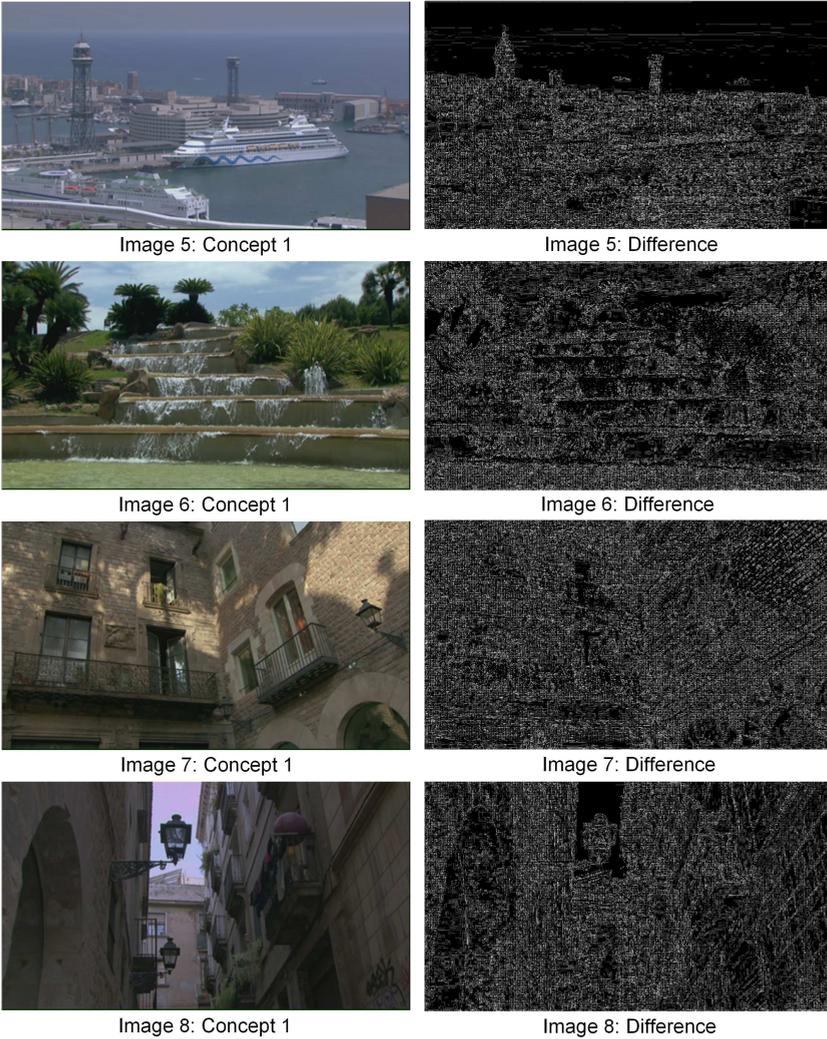


Figure 4.15 — *Final PiP image quality obtained from decimation-filtered and down-sampled intraframe-decoded full-HD images according to Concept 1, with the same meaning as Fig. 4.14.*

picture based on Concept 1. The quality indicated in this column is a relative quality of which it should be noted that the absolute quality of the PiP picture is much lower due the spatial decimation and filtering. Given the high numbers in this column, it can be concluded that the proposed Concept 1 processing

does hardly degrade the image quality compared to the full standard decoded PiP picture.

Concept 2

Table 4.6 indicates the objective picture quality for PiP-sized fast-search video navigation images, based on Concept 2 (partial spatial prediction and partial IDCT without de-blocking and decimation filtering). This simplified result of Concept 2 is obtained by comparing it with a fully normal-sized processed picture, which is afterwards scaled to a PiP-sized picture (incl. decimation etc.). The achieved quality in terms of PSNR is in the range of 23–30 dB, which is significantly lower than Concept 1. Also the average error is considerable compared to Concept 1. The question arises whether this quality is sufficient for navigation purposes. Examples of visual results obtained by Concept 2 are depicted in Fig. 4.16. The left column shows the final PiP images, while the right column depicts the error differences obtained from a comparison between PiP-sized Concept-2 images and the previously mentioned fully processed images including filtering. The error-difference signal is amplified by a factor 8 for visualization only, as the average absolute distortion varies between 3 and 8. The intensity of the error is clearly higher than with Concept 1 and the error appears strongly at the edges of objects, instead of the smooth error distribution obtained with Concept 1. Again these pictures were also visually inspected with a TV receiver, as in the previous case with Concept 1. The navigation pictures were again observed at a PiP resolution of 480×270 pixels. The qualities of these pictures seem to be attractive for navigation purposes, as they show

Table 4.6 — *Objective quality of PiP pictures derived from downsampled partially intraframe-decoded H.264/MPEG4-AVC pictures according to Concept 2.*

Video seq. (QP=28)	Avg. error	Max. error	PiP PSNR Luminance (dB)
Image 1	4.80	144	27.09
Image 2	5.39	140	27.24
Image 3	8.09	170	24.93
Image 4	9.28	176	23.11
Image 5	5.17	169	26.94
Image 6	8.48	151	24.26
Image 7	7.09	155	26.45
Image 8	3.42	132	30.38

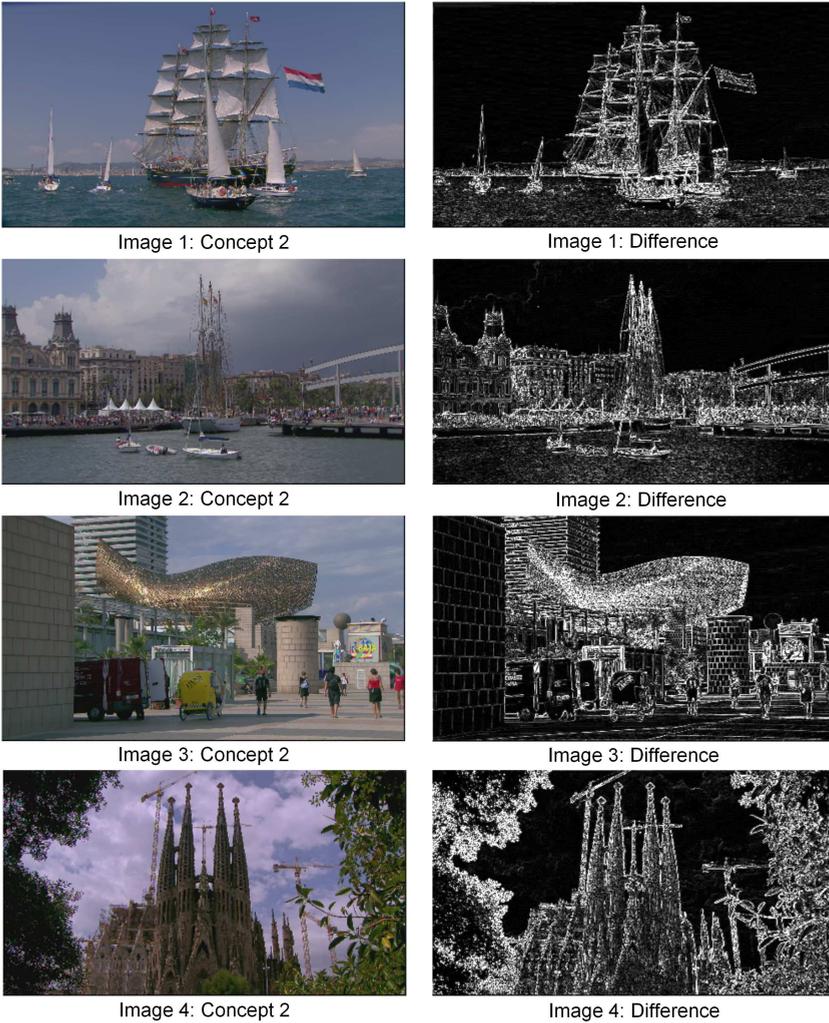


Figure 4.16 — Final PiP image quality obtained from downsampled intraframe-decoded HD images according to Concept 2. The left column shows the derived PiP images. The right column shows the difference with a PiP image obtained with full-HD decoding according to the standard followed by decimation-filtered down-sampling. For visualization, the error is amplified with a factor 8.

a certain subjective sharpness due to aliasing combined with a limited picture size. The printed pictures already indicate the presence of this subjective sharp-



Image 1: fragment full dec. PiP



Image 1: fragment partial dec. PiP



Image 2: fragment full dec. PiP

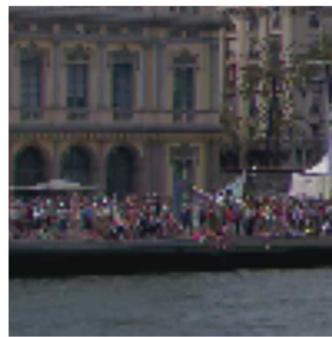


Image 2: fragment partial dec. PiP

Figure 4.17 — *Enlarged views of image parts selected from Fig. 4.16, processed with Concept 2, showing the distortion due to severe aliasing due to lack of filtering.*

ness. It is fortunate that this is acceptable because during navigation, successive images have less temporal correlation, which masks the introduced distortion and sometimes reduces the temporal fluctuations (depending on the contents). In Fig. 4.17, zoomed fragments are depicted, revealing the difference between regularly derived PiP images and PiP images computed with Concept 2. The distortions in textured areas are clearly noticeable. We have concluded that the subjective picture quality of the pictures at the right column, see Fig. 4.17, is still sufficiently good quality for fast-search video navigation, since potential artifacts are camouflaged, due to the fact that the fast-search sequence is constructed using temporally non-neighboring pictures. This causes successive navigation pictures to be less correlated, thereby masking some of the distortion.

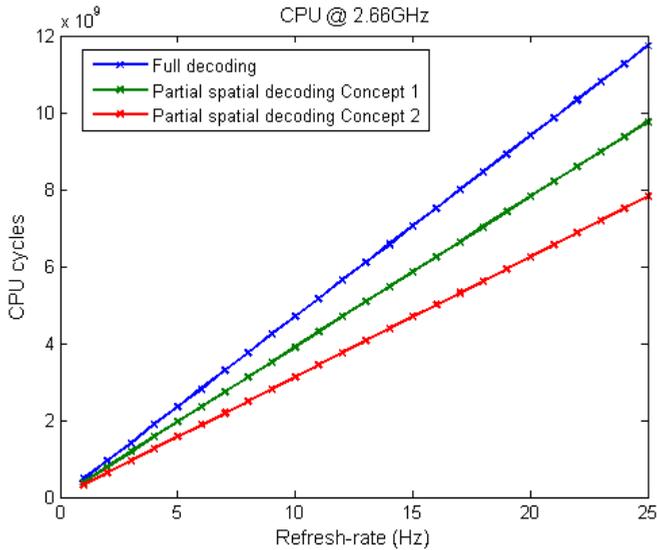


Figure 4.18 — CPU cycle load as a function of the navigation frame rate for both concepts and full decoding.

4.6.3 Computation reduction for H.264/MPEG4-AVC decoding

The previous subsection has provided the resulting picture quality associated with the proposed concepts. Let us now briefly investigate the benefit of the algorithmic simplifications in terms of cycle load. The proposed algorithmic simplifications are incorporated in the H.264/MPEG4-AVC reference software model JM18.2. The concept algorithms are executed on a PC platform to measure the performance impact. Although the platform contains a multi-core CPU (i5 core, 2.66 GHz), the program does not explore the multi-core parallelism and is executed on a single core, to mimic a single embedded CPU. The measured cycle load for the two concepts is depicted in Fig. 4.18. It can be observed that when applying Concept 1, this results in an overall cycle load reduction of 17%. When extending this into Concept 2, the computation reduction is doubled to 34%. Furthermore, Fig. 4.18 clearly shows the computational benefit when lowering the picture refresh-rate for deriving the fast-search video navigation signal. Table 4.7 shows the computation reduction for the main functions. The largest cycle-load reduction is obtained by the “memcpy” instructions, the IDCT and the absence of the de-blocking filter, whereas the cycle-load reduction for the spatial prediction block is relatively modest.

4.6.4 Test panel results on audio-enhanced dual-window navigation

The novel audio-enhanced navigation method has been subjectively tested using a small test panel consisting of 13 people. The tests are conducted using three different 25-Hz video sequences covering a news broadcast, music video and a movie sequence. The proposed video navigation sequences have been derived using the following parameters.

A relative playback speed of $P_s = 12$ has been employed for the fast-search video sequence, resulting in values for $t_{s,k}$ that are integer multiples of 0.48 seconds. The parameter T_{np} is set to 3 seconds, resulting in $t_{n,m}$ to have integer multiples of 36 seconds. These values are partly imposed by the normal-play video encoding settings, while the value of parameter T_{np} has been empirically determined. No additional processing has been applied to fine-tune the selection of the normal-play audiovisual fragments. Figure 4.19 presents a 240-ms snapshot showing a dual-window video navigation signal for the movie sequence employed in the subjective test.

The test panel members were supplied with a questionnaire containing seven questions as listed in Table 4.8. The test panel scores of the questionnaire are depicted in Table 4.9. The proposed video navigation method is well perceived, with an accumulated score for columns “good” or “better” of more than 50 % for all questions, although there are navigation aspects that have an almost equal percentage claiming the opposite. The test panel is clear on the fact that audio adds meaningful additional information during video navigation and that this method provides a global indication on the stored audiovisual content. However, the viewer does not always obtain guidance from the audio to switch between the normal-play video fragments and the PiP-based fast-search video information. Furthermore, there is also a strong variation in appreciation for the audio duration. One of the reasons for this is that the selection of the normal-play audiovisual fragments lacks additional knowledge such as scene-change detection or sound detection, resulting in a regular oc-

Table 4.7 — *Computation reduction for Concept 1 and 2 with SW-based H.264-/MPEG4-AVC intraframe decoding, divided over the main navigation functions.*

Concept	IDCT	Sp.Pred. 8×8	Sp.Pred. 4×4	Sp.Pred. 16×16	Memcpy	Remain. code	Total code
1	0%	7%	9%	25%	0%	23%	17%
2	42%	7%	9%	25%	94%	23%	34%



Figure 4.19 — Example of a snapshot of 240 ms with consecutive audio-enhanced dual-window navigation for SD video. Primary window presenting normal-play fragments, while the PiP-window shows fast-search playback with $P_s = 12$.

currence of audio intervals that contain no meaningful auditive information. Finally, the test panel indicates that an audio-enhanced double-window navigation method, does not solve the navigation efficiency problem for high-speed navigation playback.

Table 4.8 — *Questionnaire for the test panel on audio-enhanced navigation.*

Question	Text
Q1	Does the audio provide additional meaningful information?
Q2	Does the audio facilitate switching between main and PiP-window?
Q3	Does audio provide extra information when viewing the PiP-window?
Q4	Does the PiP window provide a coarse overview?
Q5	Does a 3-sec. fragment provide sufficiently detailed information?
Q6	Does the navigation provide a global impression of the AV content?
Q7	Does the navigation perform well for high speedup factors?

Table 4.9 — *Scores of test panel on the questionnaire of Table 4.8.*

Question	Test panel scores				
	Poor	Reasonable	Good	Very good	Excellent
1	0	1	6	4	2
2	2	4	6	1	0
3	0	3	8	2	0
4	1	1	5	6	0
5	0	6	2	4	1
6	0	2	6	4	1
7	0	6	3	4	0

4.7 Conclusion

In this chapter, we have proposed a video navigation method addressing the medium-time interval video navigation use case. This method employs multiple information signals to improve the perception of navigation playback. The method encompasses normal-play video fragments and corresponding audio information in combination with an additional fast-search video navigation window. The normal-play video fragments are presented in a primary window of full size, where as the fast-search video navigation is presented in a smaller secondary Picture-in-Picture (PiP) window, resulting in a dual-window video navigation solution. This navigation form can be employed as a viewing mode to summarize the stored information and offers the possibility of conducting activities in parallel to video navigation. The proposed navigation concept is not networked and requires local audiovisual decoding.

Algorithms for navigation signal processing. The algorithms for constructing the

navigation signal are as follows. The main navigation signal is based on reusing normal-play audio and video fragments of typical length of 3 seconds (approx. 6 GOPs), which are fully decoded. The fast-search navigation signal is derived from decoded normal-play intra-coded pictures, which are scalable MPEG-2 decoded already during recording. The scalability enables downscaling during decoding, since the intended picture quality is lower. These lower-quality pictures are stored, next to the other essential Characteristic Point Information (CPI) in the metadata database.

During playback navigation, this downscaled signal is recovered and decoded and mixed with the main navigation signal. When deriving an audiovisual navigation signal on the basis of concatenation of non-consecutive normal-play GOPs, the concatenated stream needs to be modified to ensure MPEG-compliant formatting and enabling seamless decoding. This modification involves amongst others audio padding, removal of predictive images without reference and modification of the time base for navigation.

Computation reduced H.264/MPEG4-AVC intraframe decoding. The usage of this advanced audio-enhanced dual-window navigation involves more complicated embedded processing that should be mapped on a consumer DTV platform. For this reason, two concepts are proposed to reduce the complexity of the embedded signal processing, particularly for H.264/MPEG4-AVC-coded signals typically used for HD television. In the first concept, the reduction is achieved by using reference pixels and repetition pixels in the calculation of the spatial prediction block. The selection of these pixels is guided by the spatial prediction patterns employed in the H.264/MPEG4-AVC standard. Also, the deblocking filter is switched-off. The second concept is an extension and on top of the previous measures, a partial IDCT is performed. In the IDCT computation, only reference pixels are reconstructed, thereby avoiding the reconstruction of residual information based on replication pixels. It was measured that in terms of cycle load of the software execution, the complexity reduction corresponding with Concept 1 and 2 was 17% and 34%, respectively. Recording of the PiP signal is typically done at 2-Hz rate only, so that the extra processing load is small.

Picture quality of scalable MPEG-2 decoding. The software simulations performing scalable MPEG-2 intraframe video decoding show a good subjective video quality. The objective quality in terms of PSNR is in the range of 26–32 dB. For video navigation, this quality is sufficient, as the viewer will have insufficient time to visually inspect the individual images. Furthermore, due to typically limited temporal correlation between successive video navigation images, individual impairments are masked.

Picture quality of computational reduced H.264/MPEG4-AVC decoding. The simplified H.264/MPEG4-AVC images have a good subjective quality for navigation purposes. When using Concept 1, the PSNR is in the range 39–46 dB, when deriving a PiP-sized picture with decimation filtering followed by subsampling. When further reducing the complexity with Concept 2, the decoded images show a considerable distortion, having an objective quality in terms of PSNR of 23–30 dB. Because of the reduced size, these PiP-sized images still have a sufficiently good subjective quality, due aliasing caused by the absence of filtering, which is partly perceived as subjective sharpness.

PVR architecture. The architecture of the proposed audio-enhanced dual-window navigation appears to be an extension of the PVR functional block diagram that was established in the previous chapter. The extra functional blocks are mainly associated with the recording processing of the PiP signal, the PiP decoding during playback and the mixing with the primary window. The main signal also requires additional audio processing such as padding adaptations, while the video processing addresses open GOPs by detection and removing predictive-coded pictures without temporal reference.

In this thesis, we have presented three video navigation solutions, each addressing a particular time interval over which the navigation is conducted. The deployed concepts show a high commonality and differ mainly in the way of presenting the video navigation information. A key aspect of this work is the re-use of normal-play encoded audiovisual information, involving specific processing in the MPEG-2 compressed domain, resulting in a compliant video navigation signal. This enables re-use of video and audio decoding components, in such a way that transcoding is avoided and that the additional processing can be embedded on the existing processing units and control CPU. Furthermore, derived CPI information is re-used by all three video navigation methods. This even holds for the navigation solutions which employ subpictures. Although the bit-cost constraints (slice level versus picture level) regarding MPEG-2 encoding differ for the mosaic-screen solution and audio-enhanced dual-window solution of video navigation, the latter can re-use the subpictures of the former for constructing the fast-search video navigation sequence.

We conclude that on the basis of the chosen concept with small-picture generation during recording and the associated metadata, the re-use of already coded pictures with scalable and/or partial decoding and the measures for complexity control, all together enable the combination of the proposed PVR concepts to create a framework that can handle short-, medium- and long-time interval video navigation. Moreover, this framework would be feasible and realizable with limited complexity.

Robustness improved DVB-H link layer

5.1 Introduction

In the fall of 2004, the European Telecommunications Standards Institute (ETSI) approved the Digital Video Broadcast Handheld (DVB-H) standard [33], [34], which is specifically tailored to battery-powered mobile reception. Basically, the DVB-H standard is an extension of the already existing DVB-T standard for terrestrial communication, but with extra features added to the physical and link layer. Although DVB-T is capable of providing mobile television reception, this standard is not efficient and robust for mobile, handheld battery-powered reception. In order to address these two system aspects, the DVB-H standard has incorporated several features on top of the terrestrial communication standard, called DVB-T. More specifically, these additional features are added to the DVB-T physical and link layer, which particularly contribute to the efficiency and robustness. This makes DVB-H a superset of DVB-T.

Let us now elaborate further on the essential features of the DVB-H link layer. The DVB-H link layer uses a Time-Division-Multiplex (TDM) broadcast technique, called time-slicing, to transmit a service, enabling power-efficient service reception and service discovery in neighboring broadcast cells (areas). Furthermore, the DVB-H link layer is equipped with a second Forward Error Correction (FEC) layer, called MPE-FEC, which forms part of the link layer, applying a [255,191,65] Reed-Solomon (RS) code. The coding distance $d = 65$ of this RS code allows to correct up to e erasures and t errors as long as the inequality $2t + e < d$ is fulfilled.

This additional MPE-FEC protects the received service against various reception impairments, e.g. Additive White Gaussian Noise (AGWN), varying channel conditions due to mobility or impulse noise influences. Unlike other DVB standards, DVB-H is an IP-based broadcast system, employing a Multi-Protocol Encapsulation (MPE) section to encapsulate a single IP datagram, or a Multi-Protocol Encapsulation Forward Error Correction (MPE-FEC) section to encapsulate RS parity data. Although the additional DVB-H link layer FEC

contributes to an improved robustness, this robustness improvement comes at the expense of duplicated IP datagram exchange between data link layer and network layer and out-of-order reception of those datagrams by the network layer, resulting in additional data processing and communication. The root cause for this additional data processing is that the DVB-H standard does not facilitate the retrieval of only correctly received IP datagrams from a *defect* MPE-FEC frame after the link layer correction (MPE-FEC frame is the data field area upon which the secondary FEC is active). As a consequence, all correctly received IP datagrams are already forwarded to the network layer, prior to the correction of whole application data table by the link layer FEC, and stored in the MPE-FEC frame (the application data table forms the memory containing the IP datagrams). When the link layer FEC has corrected all erroneously received data, all IP datagrams (correctly received and earlier erroneously received) are forwarded to the network layer, resulting in (1) data duplication of already correct IP datagrams and (2) frequently occurring out-of-order reception due to forwarding of already correct IP datagrams prior to communicating a possible fully-corrected MPE-FEC frame.

This chapter aims at solving the previously described inefficient IP datagram communication, thereby improving the associated energy consumption. Furthermore, we will propose a new algorithm to further improve the robustness of the DVB-H communication, by still employing corrected IP datagrams within a defect MPE-FEC frame. When these improvements have been established, it will be shown that both robustness and the associated Quality of Service (QoS) result in a significantly improved performance.

The sequel of this chapter is organized as follows. Section 5.2 elaborates on the standard link layer as part of the DVB-H standard. Section 5.3 introduces a novel concept for a DVB-H link layer with minimized data communication and improved QoS. Section 5.4 presents our improved DVB-H link layer, while Section 5.5 provides the corresponding experimental results. Finally, conclusions are presented in Section 5.6.

5.2 DVB-H link layer essentials

This section introduces the standard DVB-H link layer, its framework and its position in a typical DVB-H receiver system. Furthermore, the DVB-H broadcast protocol stack is briefly discussed together with specific DVB-H features.

A. DVB-H receiver system

The DVB-H standard enables IP-based service reception on mobile handheld battery-powered receivers, based on a Time-Division-Multiplexing (TDM) service broadcast. The DVB-H link layer forms the interface between the physical

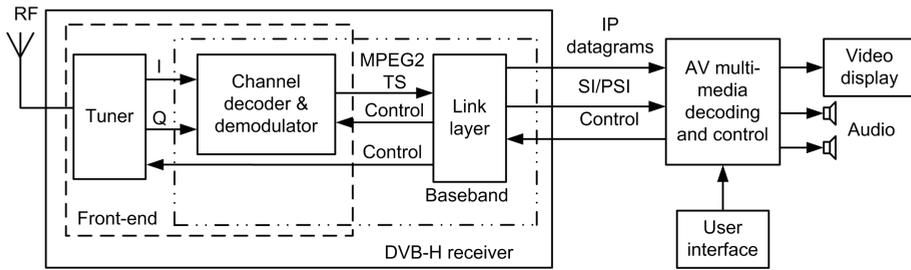


Figure 5.1 — DVB-H receiver system.

layer and the network layer, see Section 5.2.0 - C, and demultiplexes the received MPEG-2 TS, provided by the physical layer, into IP datagrams (service), Service Information (SI) and Program Specific Information (PSI), see PSI section usage in Fig. 5.4. Figure 5.1 shows a basic DVB-H receiver system, consisting of a DVB-H receiver and audiovisual multimedia decoder with control. In this DVB-H receiver, the tuner is controlled not only by the receiver application, but also by the link layer, enabling a sleep mode, which is facilitated by the TDM-based service broadcast and controlled by the link layer.

The DVB-H broadcast consists of MPE sections containing IP datagrams and MPE-FEC sections carrying Reed-Solomon (RS) parity data.

B. Multi-Protocol Encapsulation (MPE) and MPE Forward Error Correction (MPE-FEC)

DVB-H employs an MPE section [96] to transmit a single IPv4- or IPv6-based datagram. Optionally, RS parity data is transmitted using an MPE-FEC section. An MPE section for DVB-H is based on a modified private section format [36], where the *MAC_address* fields are replaced by the *real_time_parameters* fields [96]. The *real_time_parameters* fields [97] contain four parameters required by the DVB-H link layer to perform: (1) service synchronization, (2) correct storage of IP datagrams and RS parity in the MPE-FEC frame, (3) power-down control of the receiver front-end and (4) signaling the end-of-service data. The section carrying the RS parity data is called MPE-FEC section [97] and is also equipped with the *real_time_parameters* fields. The MPE-FEC sections are compliant to the DSMCC_section type "User private" [96]. The four *real_time_parameters* information fields are summarized below.

- **Address:** The address field contains the MPE-FEC frame start address position of the first Byte of the section payload, enabling an appropriate

placement of received IP datagrams in the application data table or RS parity data in the RS data table.

- **Delta.t:** The *delta.t* field indicates when the next service burst is broadcasted, enabling the receiver to power down between two service bursts.
- **Table.boundary:** The *table.boundary* flag, when set to “0x1” indicates the last section for the application data table or RS data table.
- **Frame.boundary:** The *frame.boundary* field, when set to “0x1” denotes that the current section is the last section within the current burst, which is either an MPE section without available FEC data, or an MPE-FEC section.

Figure 5.2 shows an MPE-FEC frame, with column-wise storage of IP datagrams and RS parity data, each in their own table. Compared to the height of an MPE-FEC frame, an IP datagram can be smaller, larger or equal to the MPE-FEC frame height, whereas the length of the RS parity data equals that of the MPE-FEC table height, which can be 256, 512, 768 or 1,024 rows. Because the storage start position of an IP datagram in the MPE-FEC frame does not necessarily start at a fixed position, correct placement is guaranteed by means of the *real.time.parameters* address field, even for error-prone service reception. It is essential to note that in a standard DVB-H link layer, the *real.time.parameters* address field is only used once during service reception.

C. DVB-H Protocol Stack

DVB-H is a broadcast transmission system for datagrams [98]. These datagrams are based on IPv4 or IPv6 [99], [100], or other network layer datagrams [101],

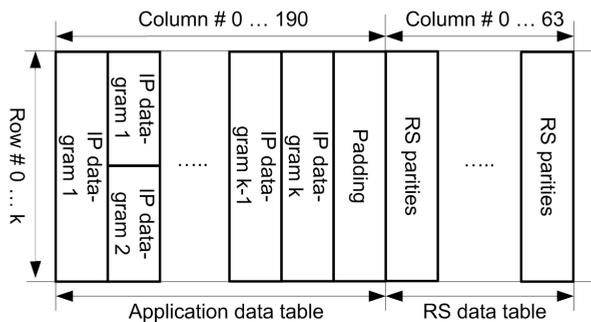


Figure 5.2 — DVB-H MPE-FEC frame. Starting from the left, the column-wise storage of IP datagrams. At the right, the column-wise storage of RS parity data.

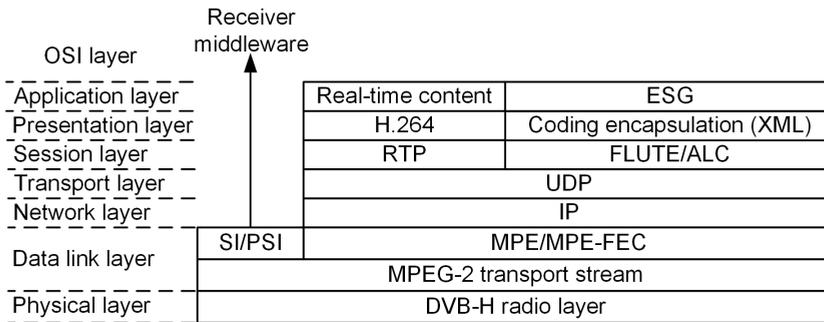


Figure 5.3 — DVB-H broadcast protocol stack.

[102] and they are encapsulated in MPE-sections [96]. Figure 5.3 depicts the DVB-H broadcast protocol stack and its relation to the OSI layering and indicates the position of the DVB-H link layer (corresponds to the same level as the OSI data link layer). The DVB-H link layer consists of IP datagrams and the MPE-FEC framing, Service Information (SI) sections and Program Specific Information (PSI) sections [103].

Figure 5.4 depicts a standard DVB-H link layer framework, enabling the filtering of multiple SI/PSI sections and IP-based services, which can be optionally protected against errors by FEC. The SI and PSI sections are forwarded to the receiver middleware, while the IP datagrams are forwarded to the network layer.

The DVB-H link layer conducts communication according to the OSI model, whereby the actual layer provides a service to the above layer, for which it delivers error-free information [104]. This error-free property of the provided information is satisfied with high probability (receivers can have occasional errors) and is ensured by two forms of error detection and correction. Error detection is enabled on the basis of a Cyclic Redundancy Check (CRC), which is part of the MPEG-2 section syntax. In a DVB-H broadcast, all data is transmitted using sections, which are protected with a CRC, mostly avoiding forwarding of corrupted data to a higher layer. Error correction is an optional DVB-H link layer feature and is only available for IP-based AV and data information.

D. DVB-H link layer framework

Figure 5.4 depicts the framework of the standard DVB-H link layer with the individual processing stages, according to the DVB-H standard [97]. At the top, an MPEG-2 TS is demultiplexed into section-based program description data

Algorithm 11 IP datagram readout

```
while no-padding do  
    version  $\leftarrow$  determine_ip_version  
    if version == IPv4  $\vee$  version == IPv6 then  
        datagram_length  $\leftarrow$  determine_IP_datagram_length  
        forward IP datagram to network layer  
        application_data_table_read_address = +datagram_length  
    end if  
end while
```

(SI/PSI) and section-based audiovisual information. Depending on the payload type, the sections are subject to either SI/PSI or MPE/MPE-FEC filtering. For the situation that the SI/PSI section CRC is correct, this section is forwarded to the middleware, or rejected otherwise. For the situation that an MPE section CRC is correct, the IP datagram is forwarded to the network layer, while the *real_time_parameters* (see right-hand side) are used in the system control. For the situation that the service is FEC protected, the IP datagram is also stored on the basis of the associated *real_time_parameters* address field in the MPE-FEC frame, enabling FEC decoding in case of reception errors during the service burst. An erroneously received IP datagram is flagged as an erasure, enabling FEC decoding. After reception of an erroneous service burst, two situations can occur: (1) FEC decoding corrects all received errors, or (2) FEC decoding fails to correct those errors. For the situation (1), the individual IP datagrams stored in the MPE-FEC frame can be retrieved according to the following essential mechanism, as shown in Algorithm 11. This readout mechanism is essentially based on parsing the IP *datagram-length* field, which is part of each datagram header. On the basis of this length field parameter, individual IP datagrams are retrieved from the MPE-FEC frame, thereby supporting services based on variable-length IP datagrams.

The problem of this simple readout mechanism is that for erroneous MPE-FEC frames, this standard DVB-H link layer cannot make a distinction between correctly received and erroneously received IP datagrams. Depending on the final error status of the MPE-FEC frame after FEC, it may be possible that no additional transmission of the corrected MPE-FEC frame is conducted, or that the MPE-FEC frame is fully forwarded to the network layer, including both the newly error-corrected packets and the already correctly received packets. As a result of this, all correctly received IP datagrams are sometimes forwarded twice to the network layer. This results in data duplication and out-of-order reception of IP datagrams, particularly when the corrected datagrams have to

The above mechanism of re-transmission effectively provides a service robustness in the communication, at the expense of extra bandwidth and thus energy consumption. When the amount of errors exceeds the error-correcting distance, the corrupted MPE-FEC frame cannot be fully corrected, resulting in the usage of only the original, correctly received datagrams. These packets construct only a part of the full data set, so that the applied data contains holes, giving distortion to the finally decoded audiovisual sequence. In conclusion, the system will work under normal operating conditions, but will immediately degrade seriously when the error-correcting distance is exceeded.

5.3 Conceptually improved DVB-H link layer

This section provides a conceptual outline for improving the robustness, while enabling a smooth signal degradation of the DVB-H link layer, aiming at minimizing the data communication.

Figure 5.5 indicates the intended impact of both proposed aspects: improved robustness and smooth signal degradation. This figure shows that when extending the 100 % data recovery interval from A to $A + B$, the robustness would be improved. Furthermore, a smooth signal degradation is obtained when the steepness of the declining curve is less strong. As a result, the reception interval between 100 % and e.g. 70 % data is extended, by offering an acceptable data recovery degree. Consequently, the potential operational interval is extended with a fraction of the worst-case intervals depicted by C and D . The curve depicted by Fig. 5.5 is a conceptual curve, which requires additional measurements to be realized on top of the DVB-H standard, in order to obtain a practical performance gain. Our proposed DVB-H link layer enhancement involves three improvement aspects: (1) higher robustness, (2) smoother degradation, attempting a better Quality-of-Service and (3) minimizing the data communication in order to reduce energy consumption.

- *Robustness.* Given the fact that the system is used in handheld operation, we expect that the received data is subject to spurious large random and small burst errors. The concept with an MPE-FEC frame with a secondary FEC enables a better exploration of the error distance provided by the FEC, as it distributes the errors. It should be utilized in a subtle way to limit and preferably remove the influence of the multiple small burst errors. This means in practice that the decoding algorithm will balance between erasure decoding for error indication and the actual error correction. This approach paves the way for a higher robustness for poor reception conditions and maximally exploiting just feasible error patterns.

- *Smooth signal degradation.* For the situation that the error pattern exceeds the FEC distance, a smooth signal degradation behavior is pursued. This behavior can be realized by fully exploiting the error-concealment possibilities enabled by the MPEG decoding algorithm [105]. Such algorithms provide a fair quality, when only 70-90% of the correct data is available. Such a concept will lead to an extra level of operation at a somewhat lower quality level, leading to a better overall Quality-of-Service (QoS).
- *Communication bandwidth.* To optimize bandwidth in a DVB-H receiver, correctly received IP datagrams are forwarded only once to the network layer. This means that the data reading process for providing data to the network layer is modified. As a result, two scenarios are supported by the implementation. The first scenario corresponds to the situation that all received IP datagrams can be successfully corrected by the FEC. In the second scenario, not all data can be corrected. In either scenario (i.e. fully corrected or partially corrected MPE-FEC frame), correct IP datagrams are forwarded only once to the network layer.

In the next section, the above aspects and directions for improvement are further elaborated and worked out in detail considering the underlying DVB-H standard.

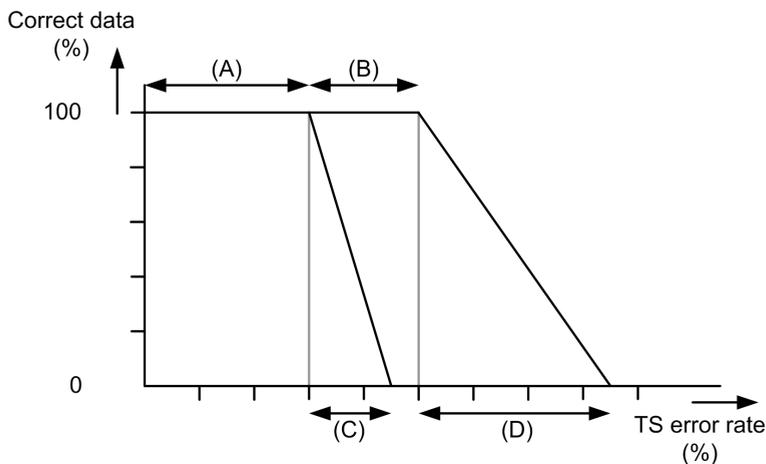


Figure 5.5 — *Improved robustness and smooth signal degradation intervals, requiring an enhanced DVB-H link layer.*

5.4 Enhanced data recovery for an improved DVB-H link layer

This section proposes an algorithm and corresponding architecture and processing stages for the implementation of the improved DVB-H link layer. The section commences with the followed approach of the solution, followed by the involved algorithms. The corresponding architecture will be presented in Section 5.5. The presented algorithms relate to improved IP datagram processing within FEC decoding using reliability information derived during service reception and FEC decoding.

5.4.1 Solution approach

The approach for our solution to improve the DVB-H link layer is achieved in the following way.

1. The received MPEG-2 TS packets contain the so-called “Transport Error Indicator (TEI)”, which indicates whether a TS packet is correctly received. This TEI parameter is employed to derive an indicator for each data byte in the TS packet payload, indicating whether that byte is reliable or not. This reliability information is kept as a side information next to the MPE-FEC frame, to be further employed by the error-correction strategy. Furthermore, the MPEG-2 TS packets also contain a so-called “Continuity Counter (CC), which can be used to reveal the absence of one or more TS packets in a sequence of TS packets. If a continuity gap occurs, the missing part of the TS payloads should be indicated as errors in the previously mentioned reliability side information.
2. The received IP datagrams are vertically stored in the MPE-FEC frame, but next to this frame, the side information with the reliability information is filled accordingly. Then, the Cyclic Redundancy Check (CRC) indicates the correctness per vertical column. This is the first step of the FEC decoding. During filling of the columns with TS payload, the start positions of correctly received IP datagrams are separately stored in a second table. This additional start position information enables recovery of correctly received IP datagrams, irrespective of their positions. This also features the potential retrieval of a set of correct bytes from the MPE-FEC frame, forming a recovered IP datagram.
3. By storing the FEC decoding result on a per row basis, the correction status of each row becomes available. This information can be combined with the previously stored reception reliability information and the location information of correctly received IP datagrams. On the basis of this combined information, the location of incorrectly received IP datagrams can be obtained and the reliability status of each IP datagram byte after

FEC decoding. This enables the retrieval of FEC-corrected IP datagrams from defect MPE-FEC frames, even after FEC decoding.

A. Visualization of the proposed concept

The above-proposed solution concept is visualized in Fig. 5.6. The picture shows the MPE-FEC frame (imposed on gray background), the two additional reliability masks (erasure at the bottom and CRIT at the left) and reliable IP datagram location mask at the top (IPET). At the top of the figure, the Internet Protocol Entry Table (IPET) stores the MPE-FEC start address of correct IP datagram, while at the right side, the Corrected Row Index Table (CRIT) stores the correctness of the FEC decoding. Below the MPE-FEC frame, the reliability information derived during service reception is stored in form of 2-bit erasure-flags, where every byte has its corresponding erasure flag. The proposed solution requires various functional extensions to the standard DVB-H link layer processing, see Fig. 5.7, where these functional extensions are highlighted in gray. At the left side of Fig. 5.7, an MPEG-2 TS enters the DVB-H link layer, which operates in a Time-Division-Multiplexing (TDM) mode on the basis of the derived δt value. During the time interval in which the service data

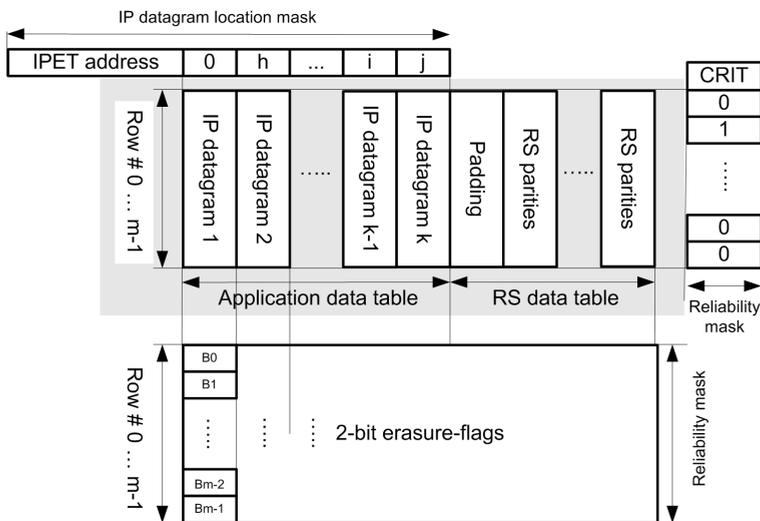


Figure 5.6 — Enhanced DVB-H link layer employing locally derived reliability information (erasure and CRIT tables) and location information for IP datagram recovery (IPET) from defect MPE-FEC frames after FEC.

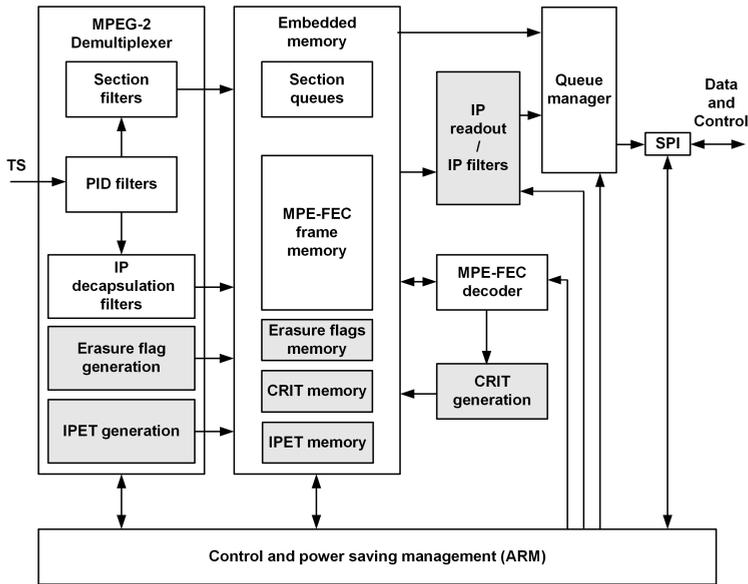


Figure 5.7 — Functional block diagram of an improved DVB-H link layer.

is transmitted, the DVB-H link layer demultiplexes the incoming MPEG-2 TS, on the basis of Packet Identifier (PID) filtering. In addition, the demultiplexed MPEG-2 TS packets are either subject to SI/PSI section filtering, or MPE/MPE-FEC section filtering. The SI/PSI filter extracts sections, which are required by the middleware and are not protected by an additional FEC, but instead are equipped with a CRC to determine correctness. The MPE/MPE-FEC filter decapsulates the IP datagram and RS parity data from the selected service burst and stores the result in the MPE-FEC frame, enabling an optional MPE-FEC decoder to correct erroneous IP datagrams. During MPE filtering, the IPET generator derives location information from the incoming data, which are stored in the IPET memory. During service burst reception, the erasure-flag generation fills the erasure-flag table with 2-bit erasure information, which is derived from the PID filtering stage.

After receiving the complete service burst, IP readout is conducted. However, in case of detected errors, the MPE-FEC decoder is invoked on each MPE-FEC row to correct erroneously received data. The CRIT generator stores on a per row basis the result of the MPE-FEC decoder in the CRIT table. For the situation that the MPE-FEC decoder corrects the complete MPE-FEC frame, IP datagram readout is conducted, which may involve an optional IP datagram filter, avoiding forwarding of undesired service data, thereby further improv-

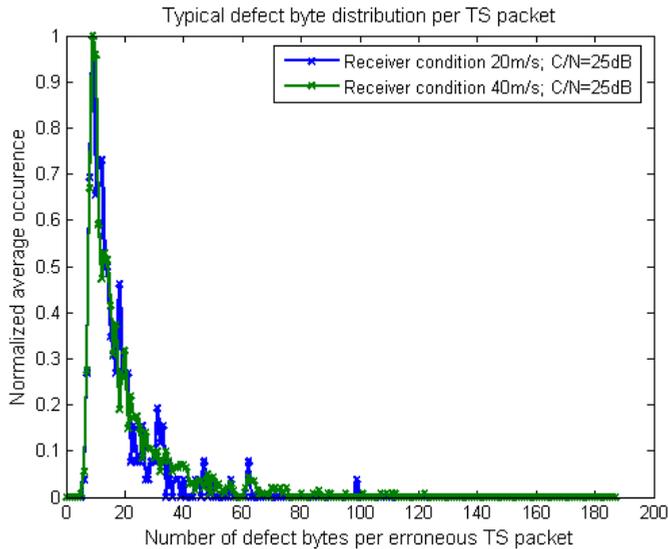


Figure 5.8 — Typical normalized number of defect bytes in a TS packet received under different conditions, where the transmission parameters are set to 16 QAM, 8k FFT and guard interval 1/8. The real probability value can be found by dividing by 9.

ing the receiver power efficiency. For the situation that the MPE-FEC decoder fails to correct an erroneously received MPE-FEC frame, IP datagram recovery is performed by the IP readout functional block by changing the reception reliability information, on the basis of the reliability information from both the CRIT and 2-bit erasure flags combined with the storage locations of the correctly received IP datagrams.

B. Validation of the channel error model

In Section 5.3, an assumption was made concerning the distribution of channel errors: multiple small burst errors versus a few long burst errors as the average behavior. This assumption is validated with a practical setup based on the official DVB-H guidelines for implementation [34]. The practical conditions are based on a Mobile Channel (TU6), with the following modulation settings: 8K FFT, Guard Interval 1/8, 16-QAM, Convolutional Code with $R=2/3$. The receiver does not use advanced Doppler compensation techniques. With these channel conditions, the corresponding error distributions have been measured of which the normalized results are shown in Fig. 5.8

The conclusion of this measurement is that the average error behavior of the channel indeed results in the occurrence of multiple small byte errors and rarely in long burst errors. A closer inspection of the figure reveals that a small amount of burst errors is indeed corrected by the channel FEC, see the left part of the curve. However, a large portion of the occurring errors cannot be corrected, which is shown by the immediate growth of the error distribution curve. Hence, the chosen Reed-Solomon (RS) FEC employed by the channel decoder, i.e. [204,188,17] RS, gives insufficient correcting performance, despite the column-row interleaving to spread the errors. As a consequence, although the DVB-H physical layer is equipped with a primary [204,188,17] RS FEC, an MPEG-2 TS packet can still be defect after channel decoding. The DVB-H standard has addressed this shortcoming by inserting a secondary FEC layer into the link layer. This means that the channel decoder first removes errors from the received TS packets, while the second RS code is utilized for removing remaining burst errors in the constructed MPE-FEC frame, as depicted in Fig. 5.6. The RS parities depicted in this figure thus refer to the second FEC processing in the link layer. The problem of this separated protection approach is that both RS codes operate independently and do not communicate with each other.

A part of our proposed solution compensates this lack of communication by adding additional reliability information derived from the individual received TS packets. This information has the form of 2-bit erasure flags to handle the reception status of the individual bytes of the IP datagrams. The bytes constructing an MPEG-2 TS packet that traveled across the typical DVB-H channel may have a different reliability status after reception and processing of the channel decoder (soft/hard erasure). Figure 5.8 reveals that a majority of the bytes may still be correct, despite the occurrence of remaining errors.

After the above explanation, we have concluded that three reception situations can occur. (1) A TS packet is received correctly. (2) A TS packet is erroneously received, indicated by the TEI flag. (3) A TS packet is lost, leading to a relatively long signal gap. These three reception situations can be distinguished with 2-bit erasure flags, which are communicated to the second FEC. The 2-bit

Table 5.1 — *Reliability information resulting in 2-bit erasure flags (TEI=transport error indicator, CC=continuity counter).*

Reception situation	erasure type	2-bit erasure flag
correct	correct	"00"
erroneous (TEI=1)	soft erased	"01"
missing/lost (CC discontinuous)	hard erased	"10"

erasure flag mapping is depicted in Table 5.1. Usage of the 2-bit erasure information as in Table 5.1 enables the distinction between correct, erroneous and missing/lost reception conditions. In the case of erroneous reception, the TEI flag is set by the channel decoder (of the PHY) and the TS payload is indicated as being unreliable (soft erased). When TS packets are missing (lost) this is detected on the basis of a discontinuity of the packet numbering indicated by the CC. The filling of the MPE-FEC frame continues with the first TS packet containing a Packet IDentifier (PID), which corresponds to the selected service PID. The absent data is labeled erroneous (hard erased). The difference between soft and hard erased is explored prior to the actual MPE-FEC decoding and is discussed in the next subsection.

5.4.2 Algorithm for IP recovery in defect MPE-FEC frame

This section presents our proposed algorithm for retrieving correctly received and FEC-corrected IP datagrams, from defect MPE-FEC frames after FEC. The algorithm is based on the reliability and location information signals as described in Section 5.4.1. IP datagrams are processed according to a three-stage approach, as indicated in Fig. 5.9. The first step is conducted during data reception, deriving reliability information. The second step is performed in case of erroneously received IP datagrams, where an attempt is made for full correction of the received errors. The third step commences when the MPE-FEC

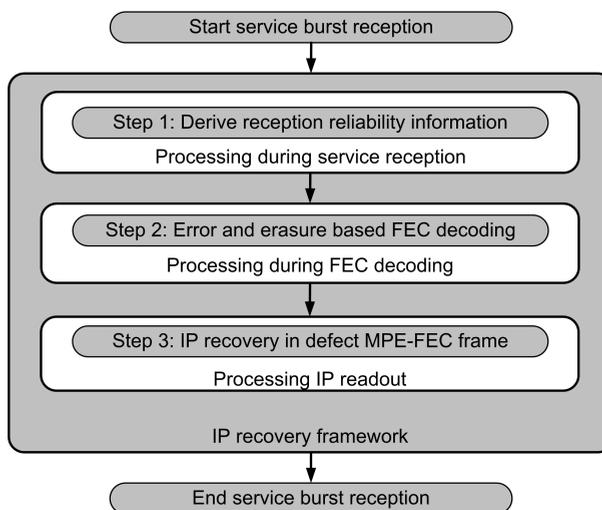


Figure 5.9 — Steps for IP datagram recovery of defect MPE-FEC frames after FEC.

frame after FEC still contains erroneous IP datagrams.

Step 1: Derive reception reliability information

Figure 5.10 depicts the flowchart showing the derivation of the reception-based reliability information. The reliability derivation processing involves readout of the MPEG-2 TS packet header (parsing) preceding the multi-protocol encapsulated data, which indicates its correctness, incorrectness or packet loss. A received TS packet is tested on correctness by evaluating the transport error indicator (in the diagram `TEI == True ?`), while the TS payload is decapsulated and the corresponding 2-bit erasure information is assigned. If preceding TS packets are lost, these lost byte positions are indicated as depicted in Table 5.1. For the situation that a correct MPE section has been received (`CRC == True ?`), the IP datagram start-address location in the MPE-FEC frame is stored in the Internet Protocol Entry Table (IPET).

For each received TS packet, with a PID equal to the service PID, the Transport Error Indicator (TEI) and the Continuity Counter (CC) are tested. For the situation that the channel decoder cannot correct an erroneously received TS packet, the TEI-flag is set to unity, leading to the 2-bit erasure flags being set to soft erased. This is motivated by the observation that still many data bytes of the TS packet can still be correct. A TS packet loss leads to a CC discontinuity and the corresponding places in the MPE-FEC frame are therefore consequently marked with a hard erasure. For the situation that an MPE section is correct, the corresponding *real_time_parameters* address is stored in IPET (see diagram `CRC == True ?`). The *PrevCC* is updated considering the current value for *PrevCC*, the CC value and the TEI flag.

Step 2: Error- and erasure-based FEC decoding

Figure 5.11 visualizes the flowchart describing the processing associated with the FEC decoding. In order to improve the DVB-H link layer robustness, a combined error and erasure decoding is applied, employing the derived 2-bit erasure flags. Prior to the actual FEC decoding, the inequality $2t + \epsilon < d$ is evaluated based on the derived 2-bit erasure information. Hereby ϵ corresponds to the sum of “soft erasures” and “hard erasures”, while $2t$ denotes the amount of errors at unknown positions with unknown value. In case the total amount of erasures exceeds the distance $d - 1$ (see diagram `sum > d-1 ?`), the soft erasures are degraded to the value “correct”. The philosophy behind this operation is that most of the soft-erased bytes might be “correct” and by lowering the number of erasures, more capacity is released for correcting possible remaining errors among the soft-erased positions. The result of the FEC calculation (see diagram `row == correct ?`) is stored in the Corrected Row Index Table (CRIT).

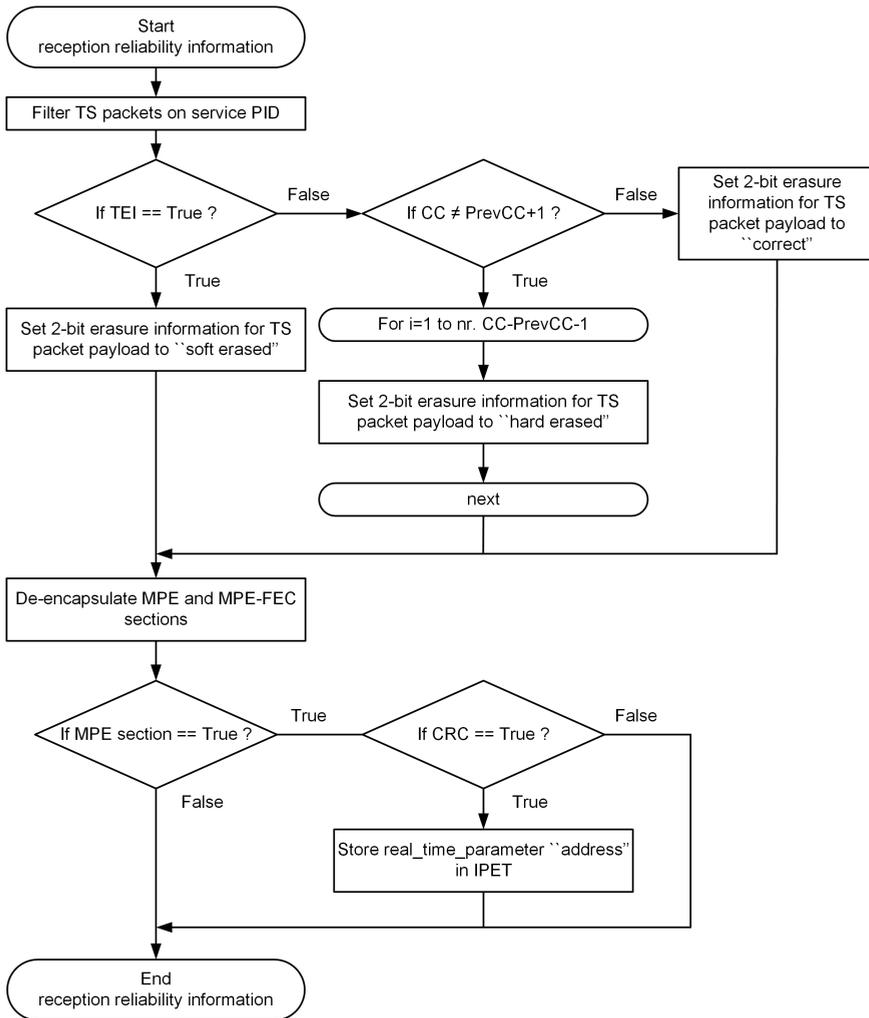


Figure 5.10 — Algorithm for generating reception-based reliability information.

Step 3: IP recovery in defect MPE-FEC frame

Figure 5.12 portrays the flowchart showing the IP recovery processing, which is applied to defect MPE-FEC frames after FEC. The correctly received IP datagrams are retrieved on the basis of their start addresses, which are contained in the IPET. Erroneously received IP datagrams result in an IPET discontinuity, which becomes apparent as there is a mismatch between the stored IP datagram

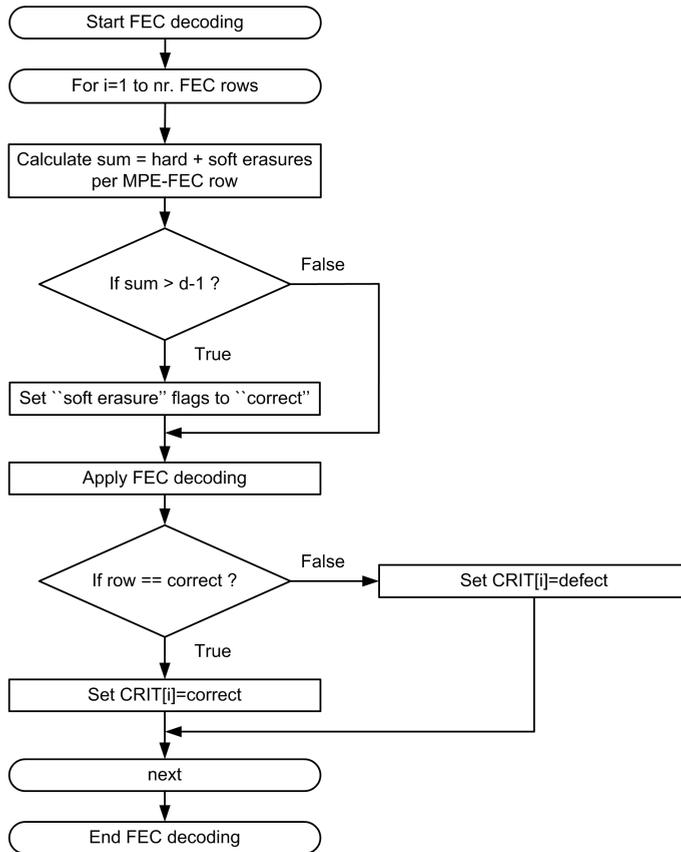


Figure 5.11 — FEC decoding based on erasure flags applying erasure degradation of “soft-erasure” flags, whereby i indicates the row index of the MPE-FEC frame.

start address and the calculated start address. This next address calculation is based on the IP datagram length field of the retrieved IP datagram (see diagram address \neq IPET ?). IP datagram(s) stored at such a discontinuity location are potentially recovered on the basis of the derived reliability information. After retrieving the last correctly received IP datagram, an attempt is made to further recover potentially corrected IP datagrams until an error occurs (see diagram while continue ?).

The next step of the required processing involves a detailed discussion on

the actual IP recovery. This is presented as follows. First the IP recovery algorithm is given in pseudo-code, representing the main body of the IP datagram recovery as shown in Fig. 5.12. The actual recovery has two separate recovery modules, the first module recovers potentially corrected IP datagrams located between correctly received IP datagrams (IP datagram recovery 1), while the second module recovers corrected IP datagrams succeeding the last correctly received IP datagram (IP datagram recovery 2).

The main body of IP datagram recovery is detailed in Algorithm 12 and described later after a short introduction on the involved processing steps. Due to the fact that correctly received IP datagrams can be preceded or succeeded by

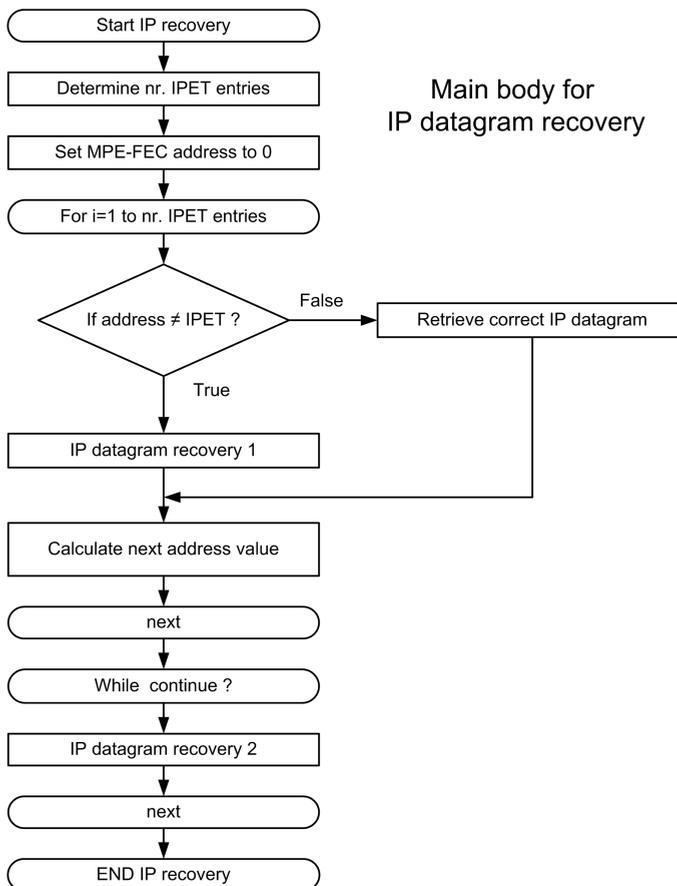


Figure 5.12 — IP datagram recovery from defect MPE-FEC frames after FEC.

Algorithm 13 IP recovery in defect MPE-FEC frame after FEC preceding a correct IP datagram

```

function IP-RECOVERY-1()
  repeat
    error = FALSE ▷ reset error signal
    for j = address to address + 6 do ▷ Check first 6 symbols
      if (CRIT[j] == 0 ∧ Erflgs[j] ≠ 0) then
        error = TRUE ▷ one or more symbols are defect
      end if
    end for
    if ¬error then ▷ first 6 symbols are correct
      read IPversion ▷ determine IP version for this IP datagram
      read IPlength ▷ for either IPv4 or IPv6 datagram
      for j = address to address + IPlength do ▷ verify symbols
        if (CRIT[j] == 0 ∧ Erflgs[j] ≠ 0) then
          error = TRUE ▷ one or more symbols are defect
        end if
      end for
      if ¬error then ▷ first all symbols are correct
        read IP datagram ▷ Recovered IP datagram
        address = +IPlength
      else
        address = +IPlength ▷ skip IP datagram
      end if
    else
      address = IPET[i] ▷ IP datagram can not be recovered
    end if
    until address == IPET[i] ▷ check for multiple IP datagrams
  end function

```

gorithm becomes active: Algorithm 13, of which the involved processing steps are now discussed. The discontinuity in the IPET address information indicates that an IP datagram is lost. In order to retrieve this IP datagram, the first 6 Bytes are tested on correctness. This test involves the corresponding CRIT value and the 2-bit erasure information (see pseudo-code “test ($CRIT[j] == 0 \wedge Erflgs[j] \neq 0$) ?”). For the situation that one or more bytes are defect, the MPE-FEC read address is made equal to the address indicated by the IPET entry, thereby skipping the recovery process. For the situation that the first 6 Bytes of the IP datagram are correct, the IP datagram type (either IPv4 or IPv6) and its corresponding length can be determined, enabling to test all bytes constructing that IP datagram. For the situation that all bytes are correct, the

Algorithm 14 IP recovery in defect MPE-FEC frame after FEC succeeding the last correct IP datagram

```

function IP-RECOVERY-2()
  repeat
    error = FALSE                                ▷ reset error signal
    for j = address to address + 6 do          ▷ Check first 6 symbols
      if (CRIT[j] == 0 ∧ Erflgs[j] ≠ 0) then    ▷ symbol error
        error = TRUE                                ▷ one or more symbols are defect
      end if
    end for
    if ¬error then                                  ▷ first 6 symbols are correct
      read IPversion                                ▷ determine IP version for this IP datagram
      if ¬(IPversion = IPv4 ∨ IPversion == IPv6) then
        error = TRUE                                ▷ No valid IP version
      else
        read IPlength                                ▷ for either IPv4 or IPv6 datagram
        for j = address to address + IPlength do  ▷ verify symbol
          if (CRIT[j] == 0 ∧ Erflgs[j] ≠ 0) then  ▷ symbol error
            error = TRUE                                ▷ one or more symbols are defect
          end if
        end for
        if ¬error then                                ▷ first all symbols are correct
          read IP datagram                            ▷ Recovered IP datagram
          address = +IPlength
        else
          address = +IPlength                        ▷ skip IP datagram
          error = FALSE                                ▷ reset error signal
        end if
      end if
    end if
    until error == TRUE ∨ address == EOT)    ▷ check for end condition
  end function

```

IP datagram is recovered, otherwise the IP datagram is discarded. This IP recovery process continues (see pseudo-code “repeat loop”) until the MPE-FEC read address equals the address indicated by the IPET entry. In this way, either none, one or more IP datagrams can be recovered.

The main body processing loop in Algorithm 12 is capable of retrieving correctly received and potentially corrected IP datagrams. After finishing these processing steps, there may still be corrected IP datagrams succeeding the last correctly received IP datagram, which requires additional processing steps for

recovery, as indicated by Algorithm 14.

This Algorithm 14 is invoked when all correctly received IP datagrams are retrieved, while the MPE-FEC frame still has remaining data positions. The processing steps of Algorithm 14 differ from Algorithm 13 due to the absence of the next entry point in the MPE-FEC frame revealing a succeeding correct IP datagram. The first step in Algorithm 14 is to determine the correctness of the first 6 Bytes, enabling the determination of the IP datagram version (see pseudo-code “test $\neg(IPversion = IPv4 \vee IPversion == IPv6) ?$ ”). In case of an invalid IP version, i.e. padding, the IP recovery process is aborted, otherwise the IP datagram length is determined, followed by the analysis of the successive IP datagram bytes. For the situation that there are one or more errors, the IP datagram is not recovered, otherwise the IP datagram is recovered. This recovery process is repeated until (see pseudo-code “test $error == TRUE \vee address == EOT ?$ ”), i.e. an error in the IP header or the End-Of-Table (EOT) has occurred indicating the end of the MPE-FEC frame.

5.5 Implementation and performance evaluation of the improved DVB-H link layer

This section presents an improved DVB-H link layer framework, suitable for deploying the improvements discussed in Section 5.4. Moreover, we discuss an elegant validation/verification concept, involving a DVB-H data generator and data analyzer. The data generator and data analyzer allow both software-based system validation and system verification involving a hardware set-up. On the basis of this validation/verification concept, results are obtained on robustness and smooth signal degradation of our proposed improved DVB-H link layer.

5.5.1 Improved DVB-H link layer framework

The involved signal processing functions as discussed in Section 5.4, are depicted in the DVB-H link layer framework diagram. This diagram is an extension on the basic decoder signal flow diagram corresponding to the DVB-H standard. This diagram was presented earlier in this chapter, see Fig. 5.4. The improvement functions are added to this diagram and indicated as gray blocks. Furthermore, this figure contains a dashed region, reflecting a strong data dependence and mutual interaction. This area represents the coupling between the primary FEC layer in the channel decoder and the secondary FEC layer indicated lower in the diagram. The new diagram with the improvement functions is shown in Fig. 5.13.

Let us now discuss the framework diagram depicted in Fig. 5.13. The de-

scription below builds further on the explanation of diagram Fig. 5.4 in Section 5.2.0 - A. After finding the TS packet header, PID filtering is applied in order to access the data streams. From the received TS packets, the section-based data is collected. Furthermore, from the TS header, the TEI and CC fields are extracted for further processing. We now enter the four proposed new functional blocks: IPET, 2-bit erasure flag generation, CRIT and the modified IP readout.

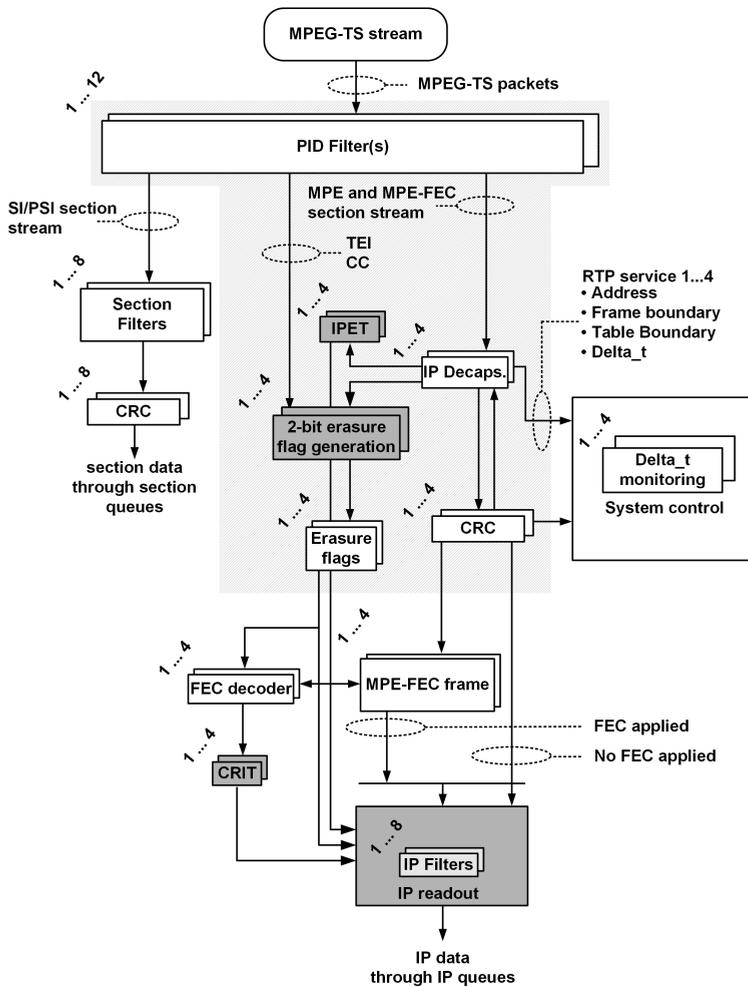


Figure 5.13 — Diagram of improved DVB-H link layer framework for SI/PSI data, FEC-based and FEC-less IP reception. The gray blocks indicate new added functions.

During TS packet reception, the TEI and CC fields are forwarded to the new block 2-bit erasure flag generation. This block utilizes these two parameters to derive the corresponding erasure flag, which are stored in the block Erasure flag. Furthermore, for each correctly received MPE-section (CRC correct), the MPE-FEC frame address field is stored in IPET. If during service reception errors are absent, the conventional IP datagram readout (IP readout) processing is applied. When errors occur, the derived erasure flags are used to support the secondary FEC layer offered by the FEC decoder block. During FEC decoding, the FEC result is stored on a per row basis in the CRIT, which is the new extension table for detecting reliability after FEC. For the situation that the secondary FEC layer corrects all erroneous data, again the conventional IP datagram readout (IP readout) processing is applied, which is based on the IP datagram length field. However, in case of uncorrectable errors, the MPE-FEC frame is considered defect in the original situation. In the improved diagram, the derived reception reliability information, location information and FEC-based reliability information are employed to recover both correctly received and corrected IP datagrams from this defect MPE-FEC frame.

In the sequel, we present the validation and verification in several steps, distributed over four subsections.

- Section 5.5.2 discusses the test set-up for software-based validation of our proposed DVB-H link layer.
- Section 5.5.3 presents the performance curves obtained by the test set-up of the preceding subsection.
- Section 5.5.4 shows the test set-up for the hardware implementation of our proposed DVB-H link layer.
- Section 5.5.5 presents the performance curves obtained by the test set-up of the preceding subsection on hardware verification.

5.5.2 Validation test set-up for DVB-H link layer

For validation of the improved DVB-H link layer, we start with an efficient concept, involving a software-based system validation approach [26]. This system validation approach is conducted for both the standard and improved DVB-H link layer, enabling performance comparison regarding robustness and smooth signal degradation. This subsection describes the validation test set-up, while the simulation results are presented in the next Section 5.5.3. The system set-up for software-based validation is depicted in Fig. 5.14 and consists of three functional blocks. The enclosed data generator provides a final, complete, compliant MPEG-2 TS test sequence. Moreover, the data generator simultaneously

produces the corresponding reference data set, in the form of an MPE-FEC frame with the automated back-annotated erasure information indicated on a per-byte basis. This concept is based on the following approach and fundamental steps.

- **DVB-H link layer reference model.** In order to avoid a full functional DVB-H link layer reference model, based on a separate encoder and decoder and a transmission channel model in between, we have used a data generator that provides the error patterns mimicking a full reference model. The generator produces next to the full functional and MPEG-2-compliant TS, a reference data set for the link layer with corresponding erasure information. We call this approach back-annotated erasure information.
- **Erasures generation.** Erasures can be generated on the basis of error traces derived from an actual service reception, or synthesized using a particular error distribution. Both methods have been adopted, whereby the method of using real captured error traces is obtained on the basis of an existing DVB-T receiver. These error traces have been obtained by employing recommended channel models of typical DVB-H use cases [34]. These models consist of urban and non-urban reception situations. For the situation that error traces are synthesized, an error distribution is employed such that the length of a burst error is limited within one TS packet and was chosen to have a fixed length of 40 bytes (equal to the measured average length in practice).
- **Worst-case erasure types.** The captured error traces have been modified by forcing “soft”-erased TS packets to be “hard”-erased, where the erasure types were explained in Section 5.4.1. This new reception condition resembles the worst-case communication situation.
- **IP decapsulation.** The back-annotated erasures depend on the applied erasure generation method employed by any DVB-H link layer, which relies on the applied IP decapsulation method. This method differs from the standard method described in [97] and presented in Section 5.4. The main

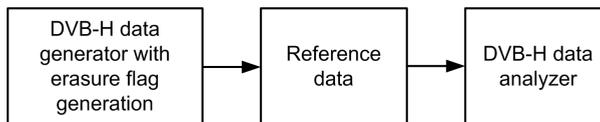


Figure 5.14 — *Test set-up for validating the DVB-H link layer performance with software implementation.*

difference between the two methods is that our method derives erasure information from the channel decoder on the basis of received or absent TS packets (see Section 5.4.1), while the standard method derives erasure information on the basis of the MPE or MPE-FEC section CRC [106].

Figure 5.14 depicts the validation set-up for both validation experiments, using a basic DVB-H link layer according to the DVB-H standard, as well as the validation set-up for an improved DVB-H link layer.

The following paragraph summarizes the operation of the actual software FEC decoder on the basis of the provided information. Based on the previously explained erasure back-annotated reference data, a software-based system validation is conducted. This back-annotated erasure information is utilized by the analyzer to calculate if an MPE-FEC frame row can be corrected by the second FEC. Therefore, the analyzer program inspects the MPE-FEC frame-based reference set data and counts the assigned erasure information. Hereby, the MPE-FEC frame is processed on a row-by-row basis. For the situation that the sum of the counted erasures satisfies inequality $2t + \epsilon < d$, an MPE-FEC frame row is correctable, otherwise the considered row remains defect. For the situation that an MPE-FEC frame could not be corrected, the analyzer can also predict the expected IP datagrams delivered by the emulated DVB-H link layer, on the basis of available reliability information added by the data generator.

It should be noticed that the implemented software-based validation is a very simple model that emulates the FEC decoding performance only. This is sufficient for algorithmic validation of the possible improvements on IP decapsulation and IP datagram recovery while concentrating on increasing robustness and smooth signal degradation.

5.5.3 Algorithm performance validation

The objective of this subsection is to compare the performance curves on the improved DVB-H link layer with the basic DVB-H link layer proposed by the standard [97]. The performance data is produced by the software-based simulation test set-up discussed in the previous subsection. The software-based validation is conducted on the basis of the following parameter settings.

- **Error model.** The DVB-H data generator produces an MPEG-2 TS with an – on the average – constant error probability. The corresponding MPE-FEC frame-based reference data is equipped with “hard” erasure information (explained earlier), revealing the defect byte positions and indicating lost data reception. This type of *erasure decoding* resembles the worst-case situation, as typically not all bytes of an erroneously received

TS packet are defect. For signal generation, the following settings have been used.

- **MPE-FEC.** The software simulation employs an MPE-FEC frame size of 1024 rows without shortening and puncturing, i.e. a code rate of 3/4. This results in a coding distance $d = 65$.
- **IP decapsulation.** All correctly or erroneously received IP data are stored in the MPE-FEC frame.
- **Service construction.** The service stream is based on a sequence of consecutive TS packets. This is the worst-case multiplexing situation, as typically services are temporally interleaved, thereby spreading the erroneous information caused by burst errors over different services.

A. Robustness and signal degradation improvements

For the basic DVB-H link layer, Fig. 5.15 depicts the simulated obtained performance curves. These curves have been constructed by employing FEC erasure decoding. This implies that the positions of the erasures are known and are derived from the TEI flag in combination with a CC error or on the basis of a CRC error indication. As a result of this analysis, the FEC decoder equation is maximized regarding error-correcting capability. In other words, the capacity for correcting random errors is sacrificed in order to have maximum capacity for correcting erasures (in $2t + \epsilon < d$, the term $2t = 0$). The following observations are made for Fig. 5.15.

- **Robustness.** Due to the CRC-based erasure flag generation, full correction of a defect MPE-FEC frame strongly depends on the actual IP datagram size. With an increased IP datagram size, more data (symbols) are lost, when employing a generic metric (like CRC) to control the erasure assignment. This explains the reason for the decreased performance for larger datagram sizes, both in robustness and signal degradation. The underlying reason for this rapid performance decay is caused by the global nature of the CRC error detection: this erasure assignment approach causes the complete IP datagram or parity data bytes to be set erased, even when correctly received. Furthermore, a rapid decrease of the robustness for larger datagram sizes is also due to the FEC parity data, which negatively influences to the overall performance. This data is also protected by means of a CRC, which causes all parity data to be erased in case of an error. As a result, a complete MPE-FEC column is erased, even if the majority of the data has been received correctly. This causes the effective coding distance d to be quickly reduced.

- **Signal degradation.** Although the robustness is limited, the curves show a smooth signal degradation for defect MPE-FEC frames, indicating the presence of information suitable for error concealment. This degradation is larger and faster for increased datagram sizes.

The previous simulation experiment has been repeated with the proposed improved DVB-H link layer. The obtained performance curves for this improved link layer are shown in Fig. 5.16. The software simulation employs the same settings as for the basic DVB-H link layer, i.e an MPE-FEC frame size of 1024 rows without shortening and puncturing. The curves depicted in Fig. 5.16 have been obtained by employing FEC *error decoding*, instead of erasure decoding in the previous experiment. With respect to the error-correction capability, only half of the amount of errors can be corrected (no erasure decoding). Due to the fact that all incorrectly received data is signaled “hard” erased (as in the previous experiment), this forms the worst-case situation. Let us now discuss the simulated link layer performance and compare this with the standard situation. The performance of the improved link layer is visualized in Fig. 5.16.

- **Robustness.** The new obtained performance curves show a significant robustness increase, providing a first drop in performance at an operation point that is 50-100% better as the standard method. This large robustness increase is caused by the fine-granular erasure assignment, enabled by the TS packet level instead of the large datagram size. As a consequence, less data are marked as hard erasure, when a small error occurs.
- **Signal degradation.** Also for the improved link layer, the signal degradation strongly depends on the employed IP datagram size. For small-sized IP datagrams smoothness is present, but this smoothness declines when increasing the IP datagram size, so that the curves become more steep. Therefore, for these worst-case simulation settings, our objective to extend the operation with smooth signal degradation is not achieved.

B. Signal degradation improvement by additional data retrieval

The signal degradation performance depicted above is obtained on the basis of correctly received and FEC-corrected IP datagrams. In the next paragraph, the simulation performance improvement is further detailed by separating the contribution of two data retrieval mechanisms. The first data retrieval is based on the correctly received datagrams, while the second retrieval mechanism is obtained by also employing the FEC-corrected datagrams in defect MPE-FEC frames after FEC decoding. The improved DVB-H link layer employs an additional table indicated by the Corrected Row Index Table (CRIT), facilitating

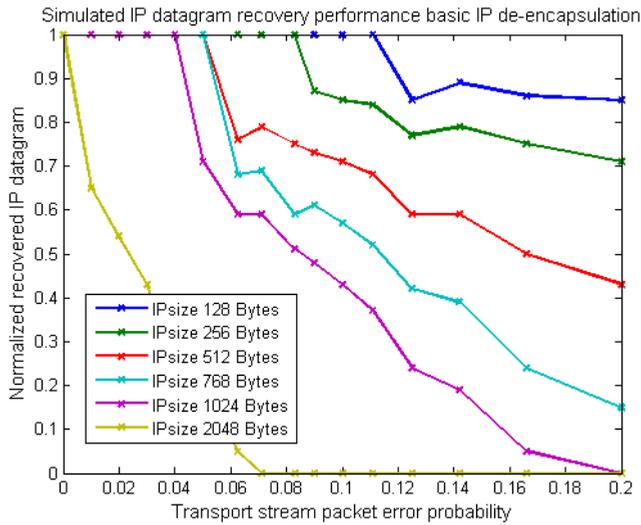


Figure 5.15 — Basic DVB-H link layer performance curves obtained by our software-based simulation model.

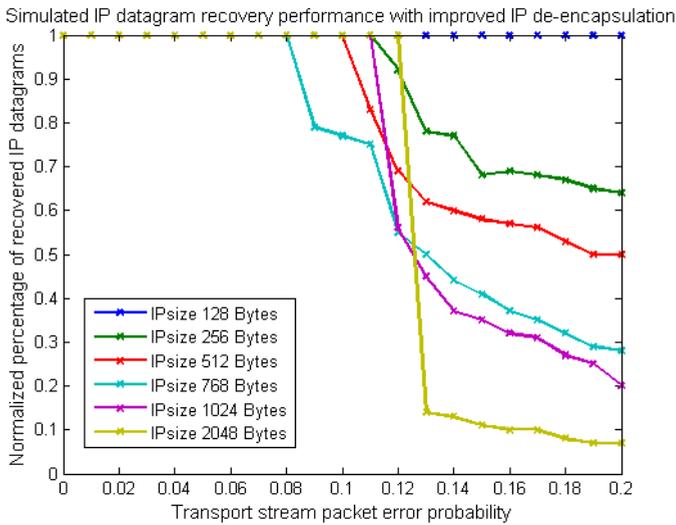


Figure 5.16 — Improved DVB-H link layer performance performance curves obtained by our software-based simulation model.

5.5. Implementation and performance evaluation of the improved DVB-H link layer

the retrieval of corrected IP datagrams in defect MPE-FEC frames after FEC decoding. Let us now elaborate on the CRIT contribution regarding IP datagram recovery. For defect MPE-FEC frames after FEC decoding, the recovered IP datagrams as depicted in Fig. 5.16, are obtained on the basis of correctly received IP datagrams indicated by IPET, and FEC-corrected IP datagrams recovered by using the CRIT table. Figure 5.17 indicates a comparison between the IP recovery performance based on using only the IPET table and the combination of location information contained by the IPET table and reliability information using the CRIT table and erasure flags, indicated as "Total". The IP recovery results for lower error probabilities are omitted in the figure, since for these error rates there is no performance difference because the FEC was able to correct all corrupted symbols, resulting in a 100 % IP datagram recovery.

- Retrieval based on combined reliability information.** IP datagram recovery based on the combined reliability information (IPET and CRIT) varies, but gives always a higher robustness and an improvement varying from a few percent up to 20 %. However, the improvements are obtained in a relatively small error probability interval of approximately 10-15% error probability. Hereby this interval strongly depends on the IP data-

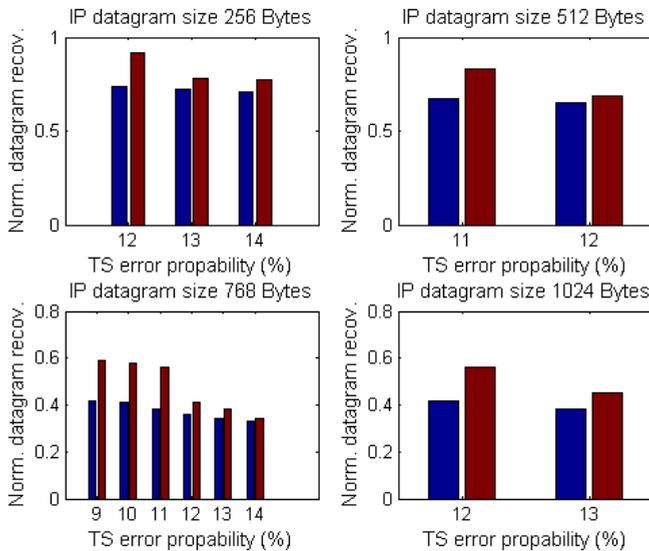


Figure 5.17 — IP datagram recovery based on IPET-only (correctly received, in blue color/left bars) versus IP datagram recovery on the basis of combined reliability information (correctly received and FEC corrected, red color/right bars).

gram size. The diagrams in Fig. 5.17 indicate that when the error probability increases, the performance improvement of the combined strategy decreases to zero, so that the performance becomes equal to the IPET-only approach, i.e. using only the correctly received IP datagrams.

5.5.4 DVB-H link layer hardware test set-up

For the verification of an improved DVB-H link layer realization with real hardware, we re-use the data generator and analyzer presented in Section 5.5.2. The verification involves next to the reference data set applied for software simulation, also an additional software-based generation of a compliant MPEG-2 transport stream (TS). This TS is generated by means of a specific software module, which is included in the DVB-H data generator. As the improved DVB-H link layer is embedded in a DVB-H receiver system, see Fig. 5.1, the MPEG-2 TS is modulated onto an RF-carrier. Although a DVB-H link layer contains various practical verification aspects, this section is limited to the verification of the IP datagram signal processing only. Figure 5.18 depicts the hardware test set-up.

Let us now describe the hardware test set-up in more detail. At the left side of Fig. 5.18, the DVB-H generator constructs an MPEG-2-compliant test sequence with or without errors, suitable for verifying the improved DVB-H link layer IP datagram processing. The MPEG-2 TS packets with possible inserted errors are modulated, enabling proper interfacing with the DVB-H receiver system in hardware, see Fig. 5.18. Prior to modulation, the channel coding adds 16 parity bytes to each MPEG-2 TS packet, thereby enabling FEC decoding by the channel decoder. As the DVB-H generator already corrupts TS-packet bytes prior to modulation, the channel FEC decoding becomes a transparent operation, as the defect bytes are handled as correct bytes. This is ensured by using a coax-cable connection between the modulator and the hardware test receiver.

At the right side of Fig. 5.18, the analyzer, which is implemented as a software program running on a personal computer, operates on two sets of information. The first set contains the IP datagrams provided by the improved DVB-H link layer, while the second set is provided by the DVB-H data generator and forms the data reference set. On the basis of the erasure information available in the reference data, the analyzer calculates the expected outcome for each received MPE-FEC frame (service burst).

The test set-up differs from the software-based validation test set-up discussed in Section 5.5.2 and applies the following different and fundamental steps.

- **IP signal processing.** Verification of the DVB-H link layer IP signal processing, involves an MPEG-2 TS in combination with the generated software reference sets. These generated reference sets employ the same con-

5.5. Implementation and performance evaluation of the improved DVB-H link layer

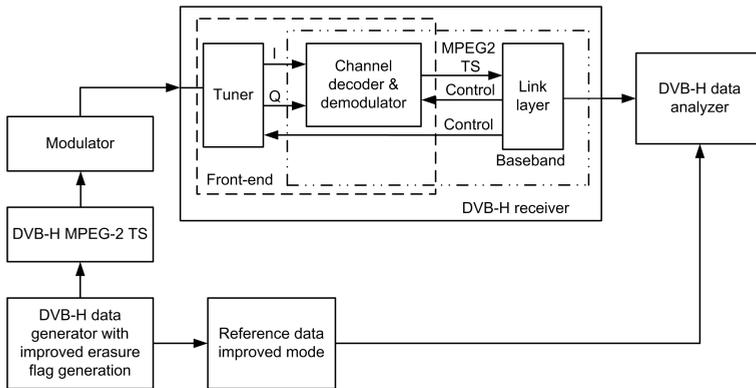


Figure 5.18 — Hardware test set-up for verification of the improved DVB-H link layer.

cept of erasure back-annotation as discussed in Section 5.5.2, enabling the data analyzer to calculate the expected IP datagrams. Furthermore, the data analyzer also receives the IP datagrams from the tested DVB-H receiver system, which is now based on an actual silicon implementation on a chip. This chip receives simulated extreme signal conditions as a test pattern. The received IP datagram information is processed and forwarded to the external data analyzer, which compares this data set against the corresponding reference set. On a per-burst basis, the analyzer provides numerical results regarding the total received IP datagrams, which can be less, equal or better than calculated by the data analyzer. This calculation depends on the applied DVB-H decapsulation method.

- **Erasures generation.** Erasures have been employed according to captured error traces derived from reception tests based on a TU-6 channel model. Thereby the following standard modulation settings have been applied (16-QAM, 8K carriers, $GI=1/4$ and $R=2/3$), which is the most appropriate modulation scheme for mobile and portable reception [34].
- **Erasures types.** The captured error traces employ both “soft”- and “hard”-erased TS packets. This enables the back-annotated erasure information to utilize both “soft”- and “hard”-erasure assignment depending on the TS packet reception status (see Section 5.4.1), similar as proposed by [107].

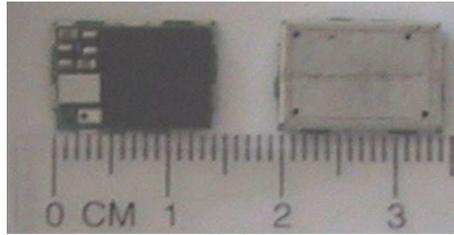


Figure 5.19 — *Multi-Chip Module of a commercial DVB-H receiver system containing the improved DVB-H link layer. Left: unshielded package, right: shielded package.*

5.5.5 Performance results of the verified improved hardware link layer

Figure 5.19 shows an actual DVB-H receiver system on the basis of a Multi-Chip Module (MCM)¹. The MCM consists of a silicon tuner and baseband chip interconnected via a laminate, packaged in a Ball Grid Array (BGA), forming a hardware implementation of the DVB-H receiver system, as depicted in Fig. 5.1. The MCM-based DVB-H receiver operates on an RF signal and delivers the requested SI, PSI and IP-based service data, according to the improved DVB-H framework as shown in Fig. 5.13. The novel MCM contains our proposed improved DVB-H link layer. This chip was exploited for verification regarding our proposed erasure flag generation and IP datagram recovery scheme.

The measured performance of IP datagram recovery leads to the following observations.

- **Robustness.** Due to the TS-based erasure flag generation, full correction of a defect MPE-FEC frame does not rely on the employed IP datagram size. Despite the fact that a DVB-H service is transmitted on the basis of consecutive TS packets and the channel introduces error bursts, with lengths of up to multiple TS packets, full MPE-FEC recovery is possible up to 10 % TS packet error rate. This significant performance improvement holds for all tested IP datagram sizes. Furthermore, all curves deteriorate at the same error probability of around 12 %.
- **Signal degradation.** The measured signal degradation strongly depends on the employed IP datagram size. For small-sized IP datagrams, a smooth

¹This MCM is a custom-made chip and commercially known as BGT205 of NXP Semiconductors.

5.5. Implementation and performance evaluation of the improved DVB-H link layer

decay of the performance curve occurs, but the smoothness declines rapidly when increasing the IP datagram size. For large-sized IP datagrams, the smoothness is negligible. Such a similar dependence was also observed during software simulation. However, the steepness of the curves is higher than the software-based simulation, so that the chip performance is lower on this aspect and resembles a typical waterfall behavior often found in digital systems.

Based on the above observations, we can conclude that the performance improvement regarding robustness predicted by the software simulation is also obtained with hardware implementation. The other performance improvement concerning the smooth signal degradation is confirmed in the hardware system. In contrast with the software-based results, the hardware system degrades more rapidly. This performance discrepancy is explained by the inequality of the error-burst distributions. Whereas in the software system the error bursts are limited to one TS packet resulting from a hard-erasure, the hardware system is exposed to error patterns which frequently exceed the TS packet length of 188 Bytes. This means that the error pattern is longer than one TS packet, leading to a higher performance degradation.

Let us now explain why the system improvement in terms of robustness and signal degradations is lower than expected. This effect is mainly caused by the

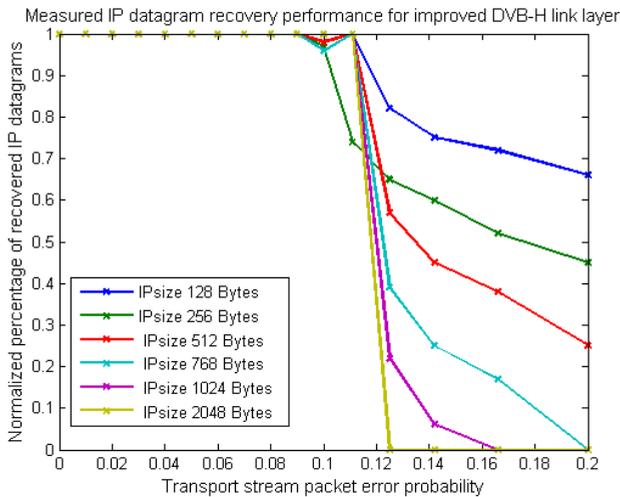


Figure 5.20 — Measured IP datagram recovery for different IP datagram sizes and TS-packet error probability, using erasure FEC decoding.

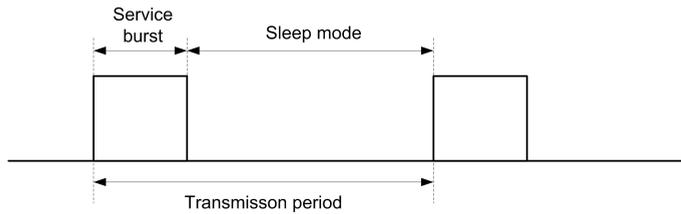


Figure 5.21 — A single transmission period of a TDM-based DVB-H service broadcast (TDM is Time Division Multiplexing).

error distribution introduced by the receiver front-end. Initially, it was expected that there would be a balance between soft- and hard-erasures, enabling erasure degradation during the second FEC stage. However, the captured error traces show that hard-erasures are dominating the defect TS packets and moreover, appear in bursts with relatively large length (10-20 TS packets). As a result of this, the error distributions are not equally distributed over the MPE-FEC frame space, which limits the full correction capability. This means effectively that the improved link layer operates with only half error correction capability in many of the reception situations.

Another remark regarding the performance is the nature of “floating” data segments caused by erroneous storage information. The term “floating” refers to data segments for which the storage position in the MPE-FEC frame is corrupted due to transmission errors, so that the decoder is not aware where to position the data segment. As a result, all segment data is lost, leading to larger chunks of erroneous symbols. With this effect, the erasure influence becomes similar to the case of the standard DVB-H link layer, giving error accumulation. This phenomenon occurs, but at a low occurrence rate so that its influence is limited.

Let us now discuss the influence of forwarding the IP datagrams only once to the network layer and its possible effect on the energy consumption. Figure 5.21 indicates the burst-based service broadcast employed by DVB-H. During reception, the receiver front-end is active and consumes typically up to 40 mW. After service reception, the receiver front-end is set into a sleep mode, reducing the energy consumption down to 1 mW. This sleep mode is entered when all correct IP datagrams have been forwarded to the network layer. The receiver shown in Fig. 5.19 has an SPI interface (see also Fig. 5.7), for exchanging data and control and operates at a 25-MHz clock. Transmitting the data contained by the MPE-FEC application table, see Fig 5.2, requires a worst-case transmission time equal to B_{size}/f_{clk} , where B_{size} denotes the maximum application data ta-

ble size in bits and $f_{clk}=25$ MHz. For the situation that the MPE-FEC frame is based on 1,024 rows, the maximum bit cost equals $191 \times 1,024 \times 8 = 1,564,672$ bits. The associated transmission time over SPI equals this amount of bits divided by 25×10^6 requiring 62.58 ms, which corresponds to 2,5 mJ additional energy consumption for each duplicated service burst. This additional energy consumption seems small, but may appear every 3 seconds, which can lead to a serious energy cost in cumulative form.

5.6 Conclusions

In this chapter, we have proposed an improved DVB-H link layer, capable of improving the robustness and providing a best-effort signal degradation, while minimizing data communication. The solution is based on locally obtained reliability and location information, in the form of 2-bit erasure flags, Internet Protocol Entry Table (IPET) and Correct Row Index Table (CRIT). The reliability information is derived on the basis of dual-stage FEC decoding, while the location information is derived from correctly received broadcast data. Hereby, the primary FEC is performed by the channel decoder, while the secondary FEC is integrated in the DVB-H link layer. The usage of the reliability information derived from the primary FEC is employed in two ways. First, this reliability information is applied for error and erasure decoding by the secondary FEC stage. For the situation that after this second FEC stage an MPE-FEC frame is still incorrect, this reliability information is used for a second time, in combination with reliability information derived from the second FEC decoding stage, supplemented with the IPET and CRIT information. In this way, correctly received and corrected IP datagrams are extracted from the defect MPE-FEC frame. Using this new link layer concept, we have found the following performance improvements.

Robustness. The usage of reliability information derived from the primary FEC to control the secondary FEC, limits the instantaneous assigned erasure information, thereby improving the second FEC stage. As a result, the robustness for retrieving completely correct MPE-FEC frames improves with approximately 50 %. Another aspect in this improvement is that the performance curves tend to cluster around the same critical performance degradation point. This tendency results in a lower dependence on the applied datagram size or parity data size, which is another benefit. In absolute sense, the dependency on the datagram size becomes also low, because the discrimination between the various curves for gradual degradation is of little relevance at the remaining performance level.

Smooth signal degradation. IP recovery in defect MPE-FEC frames on the ba-

sis of joint reliability and location information, results in up to 20 % additional IP datagram recovery. However, this performance strongly depends on the IP datagram size, as well as the error probability. As a result, there is no significant improvement on the signal degradation compared to the standard DVB-H link layer. This lack of improvement for smooth signal degradation is mainly caused by the mismatch between soft- and hard-erasures: the actual hardware generates significantly longer burst errors as initially expected, so that the expected performance gain cannot be achieved. However, this illustrates that multiple channel models apply to the real situation in practice. It may be possible for some of these models, that the degradation behavior performs more favorably. This error distribution mismatch is further discussed below.

Minimizing data communication. When forwarding IP datagrams only once to the network layer, the data communication is minimized and contributes to a reduced power consumption.

Test framework. An elegant test framework has been proposed, enabling software-based algorithm validation as well as hardware verification, avoiding the need for a full-featured DVB-H link layer reference model. By utilizing a dedicated data generator and data analyzer, early algorithm validation has been achieved, as well as full hardware verification. Elegance is obtained by employing erasure back-annotated reference data sets. These data sets contain all IP datagrams and their associated reception reliability status, thereby enabling error, erasure or erasure degradation FEC decoding emulation.

Hardware realization. A DVB-H receiver has been realized on the basis of a DVB-T baseband, extended with the specific DVB-H features combined with a silicon tuner and packaged onto a Ball Grid Array (BGA) package. The additional features encompass amongst others an adaptive equalizer for data reception, as well as hardware provisions for the improved link layer such IPET and CRIT and the associated control.

Performance differences. Software-based simulation has revealed a significant robustness improvement between the standard DVB-H link layer and our proposed DVB-H link layer. This significant robustness improvement is obtained by establishing an information link between the channel decoder FEC (in the PHY) and the DVB-H link layer FEC. In this way, the amount of unnecessary erased data is minimized, leading to an improvement of successful FEC decoding. This mechanism is responsible for the significant improvement of the robustness. This robustness improvement is also confirmed by the hardware verification.

Although the realized robustness improvement in hardware and software show a similar performance curve, it should be noted that there is a difference

between the software-based performance simulation and the hardware-based robustness improvement. The main difference between the two situations is the employed error-correcting strategy. In the hardware implementation, this error distance is fully exploited for error correction, while the software implementation employs flexible error and erasure decoding. The purpose of the software implementation was to determine the minimum improvement, while employing worst-case FEC decoding situations. In practice, the minimum improvement appeared to be also the practical improvement, because the error distribution was containing more severe burst errors, than the data generation employed for software simulation. Hence, the captured error traces are dominated by hard-erased TS packets, leaving almost no room for erasure degradation with the flexible arrangements as discussed in this chapter.

“Imagination is more important than knowledge.”

Albert Einstein, (1879 – 1955)

Block-based detection systems for visual artifact location

6.1 Introduction

Image and video communication have largely benefited from the established standards in compression techniques (e.g. JPEG/MPEG), achieved in the last decades. Many of the popular video compression techniques deploy a 2D DCT to decorrelate a block-based spatial region prior to quantization [5][108][40]. As discussed in Section 2.4, cost-effective communication removes not only irrelevant information, but also visually relevant information, thereby deteriorating the video quality.

Modern state-of-the-art LCD-based or OLED-based flat panel displays depict a high-quality picture [109][110][111], clearly revealing any video imperfections such as video coding artifacts, introduced by bandwidth-limited video communication. Digital Video Broadcasting (DVB) is based on video coding techniques, which employ a block-based Discrete Cosine Transform (DCT) to decorrelate the spatial video data. Due to quantization of the transform coefficients, artifacts such as mosquito noise, ringing, blocking and contouring are introduced. These artifacts are clearly noticeable on modern flat panel displays. However, the visibility of coding artifacts depends on the spatial complexity of the picture. Typically, coding artifacts are clearly noticeable when occurring in “flat and/or low-frequency” regions, whereas they are being masked when occurring in “textured” regions.

A global outline of a digital television receiver and the involved video processing is depicted in Fig. 6.1. Part of the video chain is the video pipe, which is a composition of video-specific processing blocks. Hereby, each individual block performs a particular video function, which can be analysis, e.g. noise meter or bandwidth meter, or enhancement such as video sharpness enhancement. In this video pipe, the embedded artifact-reduction function supports the reduction of blockiness and ringing/mosquito noise. Modern storage systems also employ image processing during playback, applying e.g. video scaling

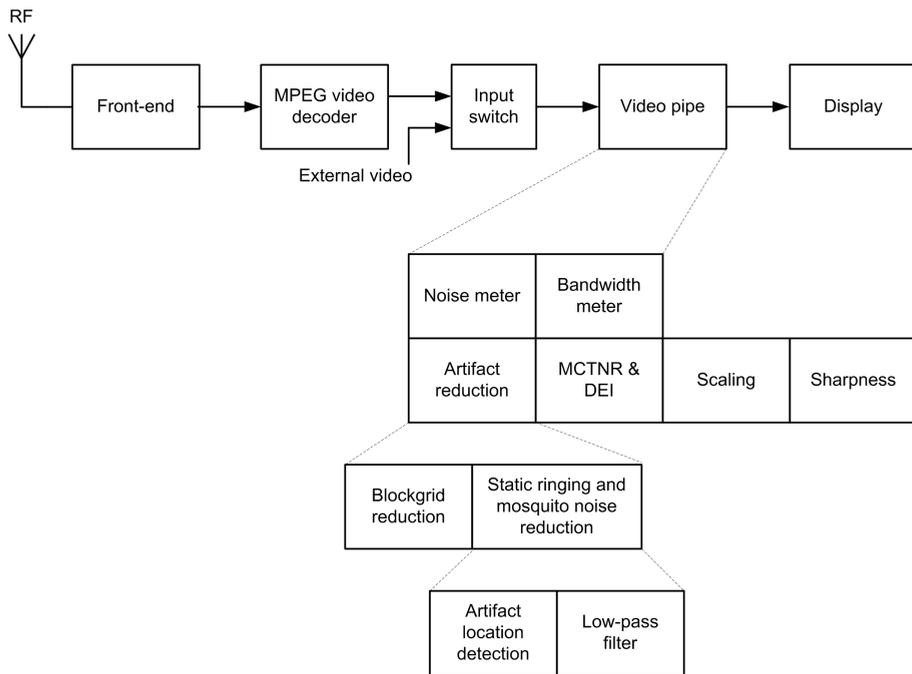


Figure 6.1 — Typical video processing chain from a digital television receiver, containing video signal improvement functions. (MCTNR: motion-compensated temporal noise reduction, DEI: de-interlacing).

and enhancement functions, which means that an external video signal as indicated in Fig. 6.1 may be an HD-like signal, originating from an upscaled SD signal. In a video sequence, coding artifacts can appear statically, dynamically or in mixed form, requiring a different artifact-reduction approach. The video chain of modern digital TV receivers employs video enhancement functions to improve the subjective picture quality after video decoding for an improved presentation on a display. This picture enhancement processing has evolved from global to local processing and from static setting to adaptive processing. This adaptivity is considering the local content of the video signal and also depends on general factors such as local contrast. In order to attenuate the visibility of coding artifacts, low-pass filtering is typically applied in a locally-adaptive manner. For example, H.264/MPEG4-AVC employs in-loop low-pass filtering in order to reduce block coding artifacts. Unfortunately, the mostly applied video communication standards lack such an internal filtering solution. Therefore, video post-processing is required in order to attenuate introduced

video coding artifacts at the receiver side. This type of video post-processing is a necessary add-on to the minimum required video processing chain. For this reason, such a sub-system is highly optimized with respect to costs. This will limit our choices for the algorithm exploration in this chapter.

Video coding artifacts can appear in different forms and can have a static or dynamic behavior. In still images, the following artifacts may be present: blocking, mosquito noise, ringing and contouring. In motion video, the previous coding artifacts may appear mostly in dynamic form, except for non-moving areas. Besides this, due to motion-compensation coding, artifacts may appear within moving objects in the form of block noise, low-frequency blocking patterns and coding noise propagation in the temporal domain, due to predictive coding of distortion occurring in reference pictures.

The following paragraphs briefly discuss typical solutions from the past two decades, which have been proposed for the three mostly occurring coding artifacts appearing in broadcast quality video.

- **Detection and reduction of visible block-grid coding artifacts:** The basic method for the reduction of blockiness employs a two-step approach, involving (1) block-grid location detection and (2) low-pass filtering of the video data at the identified location [112]–[115]. The applied methods determine the pixel locations, which are subjected to low-pass filtering to attenuate the detected block-grid boundaries. In a more advanced approach, the blockiness visibility is considered prior to final low-pass filtering [116].
- **Reduction of dynamic coding artifacts:** An alternative degradation is mosquito noise, which can be dynamic in its appearance. Solutions proposed to reduce such noise involve Temporal Noise Reduction (TNR) or Motion-Compensated Temporal Noise Reduction (MCTNR), which not only reduces dominant Gaussian noise, but up to a certain extent, also attenuates the visibility of non-static coding artifacts [117][44][118]. However, video coding artifacts are most annoying when occurring in still or slowly moving video. When objects are subject to a slow-motion speed, the viewer is able to track the objects, thereby observing the object details including visible coding artifacts.
- **Detection and reduction of visible ringing and mosquito noise coding artifacts:** Mosquito noise can also appear in static form. The methods [45], [119]–[122] for attenuating this type of artifact follows a similar two-step approach as indicated above and in Fig. 6.1. In more recent work, the Human Visual System (HVS) is employed for determining locations where the ringing is mostly visible [123], while in [124] even the amount of ringing is estimated.

The work conducted in the field of static ringing and mosquito noise detection and the corresponding attenuation [117][119][120][45][121][115] suffer from undesired image blur, which occurs when the located coding artifacts are too much suppressed. Other solutions [123][124] provide good results, but are costly in terms of computational complexity, or involve a substantial amount of embedded memory [121].

The focus in this chapter is on the trade-off between the accuracy required for finding the artifacts within the image, i.e. the detection, and the corresponding complexity for the artifact removal and/or suppression. Since the design of good filters for artifact reduction is known, we focus particularly on the detection of the artifacts and the involved complexity. To this end, two block-based artifact-location detection systems are presented, either operating in the spatial domain or frequency domain, suitable for locating the regions which potentially contain visual noticeable mosquito noise and/or ringing artifacts. The derived artifact-location information forms a control map for artifact filtering, which controls the final filtering strength of a locally-adaptive low-pass filter, as presented in [115]. The previously cited filter design will be used as a starting point for the design of the complete system and will be applied as the second stage in our artifact-reduction system. Based on this two-step approach, we aim at accurate artifact-location detection, such that visual mosquito noise and ringing artifacts are effectively removed by the adaptively controlled filter, so that the sharpness of object edges is preserved and excessive image blur is avoided as much as possible.

This chapter is organized as follows. First, Section 6.2 provides a brief introduction to mosquito noise and ringing coding artifacts. Section 6.3 proposes a framework suitable for the accurate detection of visible mosquito noise and ringing coding artifacts. Section 6.4 presents two block-based algorithms, each operating in a different domain, for detecting visible mosquito noise and ringing coding artifacts. In Section 6.5 the experimental results for both detection approaches are provided. Finally, conclusions are discussed in Section 6.6.

6.2 Background and related work

This section provides a short overview on the occurrence of mosquito noise and ringing coding artifacts. Furthermore, a few techniques are summarized for finding the locations in the image where the typical coding artifacts, such as ringing and mosquito noise, do appear. This section finishes with a summary of the requirements for blind detection of ringing and mosquito noise artifacts.

A. Background on mosquito noise and ringing coding artifacts

Various publications on ringing and mosquito noise confirm that both forms of distortion occur near the object boundaries, whereby some authors indicate

that this form of degradation is particularly noticeable in “flat and/or low-frequency” regions [117][43][125]. Some authors define the visual occurrence to be constrained, in the sense that the decoded blocks showing mosquito artifacts are based on both flat and textured areas [43]. Furthermore, another aspect is that the intensity of the ringing and mosquito noise is low, which causes this type of artifact to be masked by textured regions, while being visible in “flat and/or low-frequency” regions [117][43]. Figure 6.2 provides two visual examples of the previous observations regarding noise pattern visibility. This figure shows the original image and its decoded version after MPEG-2 compression with a ($Q = 20$). Figure 6.2(b) clearly reveals mosquito noise, which is particularly noticeable in the sky around the tree structure. Figure 6.2(d) presents an example of contamination by ringing. Again, the image degradation is clearly

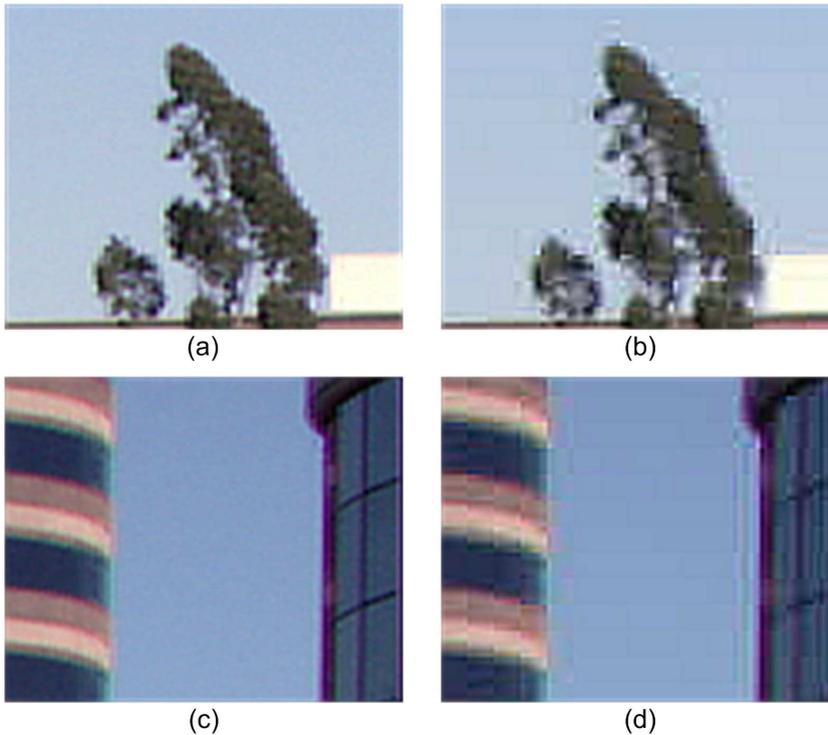


Figure 6.2 — Example of a factor-2 zoomed mosquito noise and ringing, due to MPEG-2 8×8 transform coding ($Q = 20$). (a) Original fragment. (b) Mosquito-noise fragment. (c) Original fragment. (d) Ringing fragment.

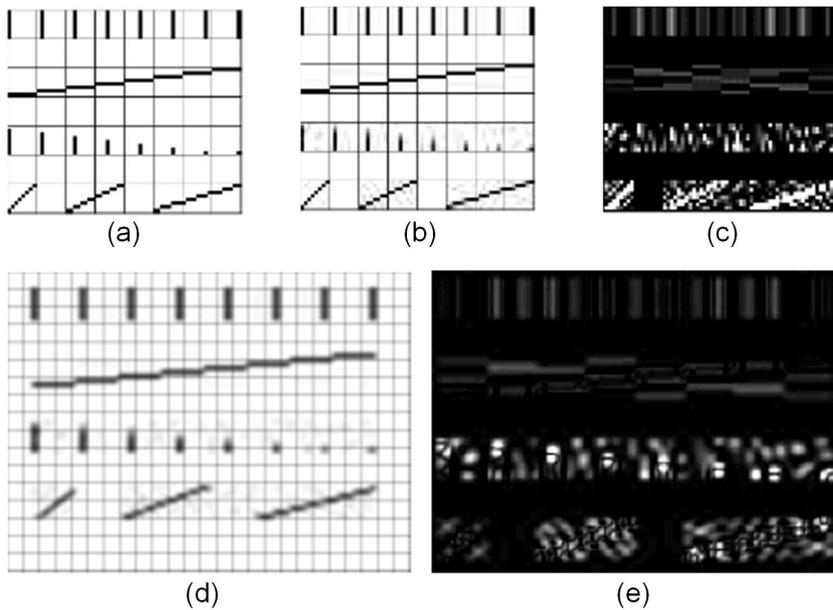


Figure 6.3 — Example of mosquito noise and ringing due to 8×8 transform coding $Q = 40$, grid size is 8×8 pixels. (a) Original video fragment. (b) Decoded video. (c) Decoding error with gain factor 8. (d) Fragment from upscaled SD (to HD) video (1920×1080 pixels), with the same grid size. (e) Decoding error for upscaled SD video fragment with gain factor 8.

visible in the sky region near the building edges. Both noise examples show that ringing and mosquito noise are visible in “flat and/or low-frequency” regions, while being masked in detailed regions. Furthermore, the noise pattern is bounded within the block size used for coding.

Let us now further analyze the emerging noise patterns when quantizing 2D DCT high-frequency components. For the typical situation that the transform block size is 8×8 pixels, the mosquito noise can occur anywhere within the 8×8 block. This can introduce a noise pattern, which can stretch itself within the pixel block up to 7 pixels in either direction, see Fig. 6.3(b)(c). Ringing occurs when the boundary of an object or a line segment of an object crosses the border of the transform block in either horizontal or vertical direction, see Fig. 6.3(b). The ringing effect is best visible in Fig. 6.3(c), which is particularly noticeable at the top (two) noise patterns of the block. The bottom (two) noise patterns reveal the mosquito noise, giving pixel noise within the block. Evidently, the ringing

location depends on the location of the horizontal or vertical object border in the transform block. The ringing effect is introduced by the quantization (attenuation or even elimination) of high-frequency DCT coefficients during compression, and is theoretically explained by the Gibbs phenomenon [35]. Both forms of distortion are spread over a larger area, when the video is upscaled to a larger image format (e.g. SD to HD conversion), see Fig. 6.3(d)(e). Note that depending on the upscaling filters, the distortion appearance may be softened, resulting in distortion blobs rather than the pixel-based noisy patterns.

Summarizing, the discussion and analysis of the decoded images lead to a few visibility rules for visible artifact detection.

- Visible and annoying appearance of mosquito noise and ringing occurs in a “flat and/or low-frequency” region near an edge/texture region transition or vice versa.
- The noise pattern stretches itself through the whole coding block.
- The visibility of the noise pattern becomes particularly annoying when the “flat and/or low-frequency” region covers a substantial area of at least several adjacent blocks.

The presented rules require that an artifact-detection kernel has means to validate these rules by measuring the conditions of the signal like local activity, and the local regions surrounding the possible noise patterns. This conclusion has been employed as a guideline for the design of the proposed detection systems presented in this chapter.

Let us now summarize the main techniques for detecting and reducing visible ringing and mosquito noise coding artifacts.

B. Related work on mosquito noise and ringing artifact reduction

Detection and associated reduction of visible ringing and mosquito noise is typically employed as part of the video processing chain in a television. Since these artifacts are dominant around edge transitions, the detection of these edges is attractive for localizing this contamination. Suitable edge detection techniques can be conducted in either the spatial domain or frequency domain, while locally-adaptive low-pass filtering is typically performed in the spatial domain.

- **Edge detection in the spatial domain:** Transition detection in the spatial domain is conducted with the luminance video information and involves typically a Sobel or Canny edge detector, or employs image analysis based on gradient information or a statistical metric [117][126][122]. In the vicinity of the detected transitions, block-based video classification

is conducted, employing features such as “texture region” and “smooth region”, revealing the video structure adjacent of the detected edges. On the basis of the calculated feature information, a binary signal is derived, indicating the coding-artifact contaminated locations, which is used to control the final low-pass filtering. Although the derivation of edge locations based on these methods typically provides good results, the overall performance of the coding artifact reduction is limited, as these methods lack the discrimination between intended texture and artifact contamination. The performance of the visible artifact-location detection can be described by the *detection score*, which to a large extent determines the artifact-reduction performance. In order to improve the *detection score*, also the chrominance video information is utilized, thereby exploring the differences in processing during video coding of the luminance and chrominance video information signal [127]. Besides classical edge detection based on a Sobel or Canny-based edge detector, more advanced solutions involve additional signal processing, while considering the Human Visual System (HVS) [123]. The performance of such an advanced proposal in terms of *detection score* is considerably increased, at the expense of a higher computational complexity.

- **Edge detection in the frequency domain:** Edge detection in the frequency domain is possible on the basis of the 8×8 2D DCT coefficients with good contour tracking and edge detection. This approach is better compared to a conventional Sobel-based detection in the spatial domain [128]. However, such an approach is expensive in terms of involved line memories, which makes this approach less attractive.
- **Low-pass filtering:** Reduction of visible coding artifacts in the vicinity of an edge, requires an edge-preserving low-pass filter. In the past decade, multiple implementations for artifact-reduction filters have been presented [45][115]. Therefore, we focus further on a good system for coding artifact detection.

We now summarize the requirements for blind detection of ringing and mosquito noise artifacts. With blind detection, we mean that there is no reference information about the original signal prior to the broadcasting signal. The quality of the resulting artifact reduction will strongly rely on the performance of the artifact-detection system. For this reason, our emphasis will be on the design of a good detection system, which is suitable for embedding in a TV processing chain, with the associated cost constraints. Further system and quality aspects and requirements of the artifact-detection system are listed and discussed below.

1. *Single component analysis*: The existing solutions for artifact-location detection are based on one signal component, the luminance. We adopt this choice, as it contains all important information on texture and edges. This also aids in constraining the involved costs.
2. *Required activity metric*: The system requires an activity metric revealing the important locations for artifact occurrence. A good metric must be edge and texture sensitive and suitable for signal characterization modeling. In this way, not only “texture” and “smooth” regions can be classified, but the metric also enables the classification of “potential ringing” and “potential mosquito noise”, thereby facilitating the construction of a simplified local signal model. Hence, after the activity measurement, the calculated activity is classified into particular signal features as above.
3. *Detection domain*: The detection of coding artifacts locations can be conducted in the spatial or frequency domain. Both approaches have their advantages so that they will be both explored.
4. *Detection aperture*: In order to reliably detect contaminated regions in the video image, the detection aperture must be sufficient to contain artifact-free background information of the video scene.
5. *Detector output*: The final detector output signal will be used as a control signal for the adaptive low-pass filter.
6. *Avoidance of switching artifacts*: The filtering stage can introduce visible artifacts in the form of on-off switching of the filter. Such artifacts are very annoying and should be avoided. An appropriate way to solve this issue, is the generation of a smooth, diamond-shaped control signal for the filter.
7. *Predetermined low-pass filter system*: As indicated in the previous section, the low-pass filter will be adopted from previous work [115]. This filter is equipped with means to control the filter settings, which can be elegantly exploited for filtering and quality control.

Based on the above requirements and system aspects, the next section presents a conceptual solution for the desired artifact detection- and reduction system.

6.3 Conceptual artifact-location detection and filtering solution

This section provides the overview of the detection system and the main system aspects. In Section 6.4, two approaches will be elaborated in more detail, one system for the spatial domain and one for the frequency domain. The low-pass

filtering applied after detection will also be explained in Section 6.4.

Based on the previously discussed visibility rules for artifact detection, we present our conceptual noise-pattern detection. The detection system incorporates two important elements: (a) a technique for measuring the activity in blocks and (b) sufficient memory for analyzing a spatial area surrounding the noise pattern. Prior to presenting the overview of actual detection, we briefly elaborate on the spatial region employed for inspecting the visibility rules. This region is implemented by a detection kernel constructed from several blocks, arranged in a two-dimensional rectangular area surrounding the actually considered block. Furthermore, each individual block of the detection kernel is evaluated with respect to its spatial activity (e.g. flat or low-frequency / texture). Figure 6.4 depicts such an artifact-detection kernel constructed from blocks. The kernel has in the center the actual block (H) for which artifact occurrence is determined. This determination is based on both local activity measurements and the resulting activity pattern that is generated throughout the kernel area, where for each block the activity is measured and classified. The two-dimensional activity pattern conceptually forms a basic model of the local video image surrounding the center block. In this way, the conditions for the visibility of the noise can be verified, particularly the condition that the “flat and/or low-frequency” region in the kernel covers a substantial area of at least several adjacent blocks. To analyze the activity pattern in the kernel in a structured way, each block is labeled with a character (A,B,...,N,O). In Figure 6.4(b), as an example, block H is contaminated due to the presence of “texture” in block D, which is visible due to the fact that block H is part of a larger “flat and/or low-frequency” region (K,L,M,N,O).

Let us now present the full diagram of our block-based artifact-location detection. Figure 6.5 depicts a block diagram of the coding artifact detection and

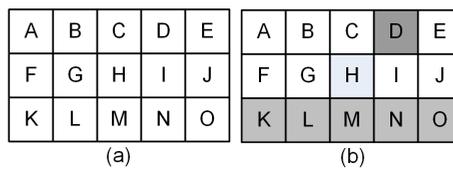


Figure 6.4 — Ringing and/or mosquito noise detection kernel. (a) Construction of a detection kernel composed of a 2D set of blocks. (b) Analysis of occurring noise patterns. Block H is located between a “textured” block D and “flat and/or low-frequency” region blocks K,L,M,N,O, which may result in visible noise.

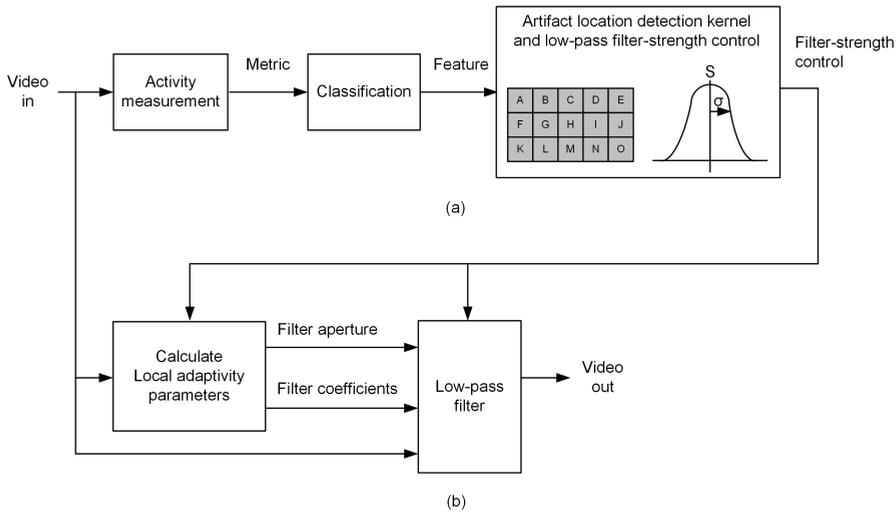


Figure 6.5 — Proposed block-based artifact-location detection system and associated low-pass filter-strength control. (a) Detection subsystem. (b) Filtering subsystem.

its connection to the low-pass filter stage. This system is based on the requirements as presented in Section 6.2. Hereby Fig. 6.5(a) illustrates the involved coding artifact detection, while Fig. 6.5(b) shows the edge-preserving low-pass filter. The proposed system allows coding artifact detection to be conducted in either the spatial or the frequency domain, involving a block-based activity metric revealing the local video variations. This exploration for both the spatial and frequency domain will be implemented and evaluated. As the applied detection kernel is composed from multiple blocks, the surrounding information of the measured video variations can also be exploited. This approach is clearly different from existing solutions in literature and will enable the system to distinguish edge information, local background information and information surrounding an edge.

The *activity metric* stage calculates the activity metric on a per block basis. The *classification* stage partitions the calculated activity metric, employing preferably discriminative features which allows local modeling of the video signal. The *artifact-location detection* stage applies a simplified model of the local video signal by classifying the activity of each block and using this in a range of surrounding blocks as input to local neighborhood analysis of the signal. The presence of potential coding artifacts, which is indicated as a binary decision, is derived for center block H of the detection kernel. From the binary-based location signal, a diamond-shaped low-pass filter control signal is derived in

the kernel. The *low-pass filter* stage reduces the coding artifacts on the basis of the calculated filter coefficients, the aperture and the diamond-shaped filter-strength signal. Let us now briefly discuss the main stages of the block diagram.

A. Activity measurement

The *activity metric* stage calculates the activity metric on a per block basis. The metric is based on the local variability of the video signal and is calculated as a 2D Sum of Absolute Differences (SAD). Since the SAD is based on differences, it is capable of finding edges, texture variation and flat background blocks. Hence, this metric is capable of classifying activity in blocks. Texture variation can also be measured in the frequency domain using a block-based transform. This is explored as an alternative.

B. Classification

The classification stage partitions the calculated block-based metric into intervals to discriminate the contents of the block. The intervals are ranked into the following classification: (1) “flat and/or low-frequency” region, (2) “potentially artifact contaminated” region and (3) “texture” region.

C. Artifact-location detection kernel

The size of the 2D detection region, i.e. the local neighborhood surrounding the center block or pixel, influences the artifact-detection accuracy and should be large enough to cover the discrimination of edges, background information and information surrounding an edge. This explains why the kernel size is chosen as a local neighborhood of 5×3 blocks, as depicted in Fig. 6.6(b). Due to the fact that ringing or mosquito noise cannot be uniquely distinguished from low-intensity texture with a simple metric, a method is required that sufficiently supports the finding of mosquito noise patterns. The block size of the employed blocks in the kernel have a size, which is smaller than the MPEG-coded transform blocks, because this provides a better detection coverage. Hence, an artifact-contaminated region in the spatial domain is defined as either a single pixel $p(x, y)$ or a group of pixels in block H, or all pixels constructing block H in a frequency-domain approach (discussed later). In either situation, a binary signal is derived as output, revealing the presence or absence of coding noise. On the basis of this binary signal, a diamond-shaped low-pass filter control signal is derived, utilizing the block features, surrounding center block H.

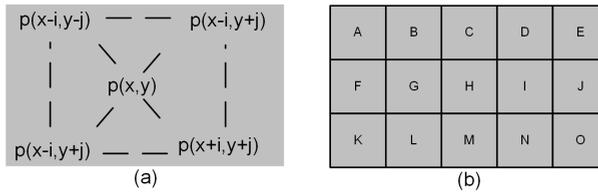


Figure 6.6 — Configuration of an artifact-location detection kernel. (a) 2D region forming a block for activity metric calculation. (b) Blocks constructing the detection kernel considering the local neighborhood.

D. Low-pass filter-strength control signal

The purpose of indicating artifact locations, is to attenuate the coding artifacts at the appropriate places, i.e. where they are typically visible. For these locations, the noise should be removed with an edge-preserving low-pass filter. In order to avoid visible switching artifacts introduced by the low-pass filter operation, we ensure a smooth filter status transition for both gradual on- and off-switching of the filter. This smoothness is obtained by a transient function that gradually increases or decreases the filtering strength. In the next section, the above algorithmic aspects are further elaborated and worked out in detail.

6.4 Block-based artifact-location detection algorithms and low-pass filter control

In this section, we propose two block-based artifact-detection algorithms, either operating in the spatial or frequency domain, suitable for detecting potential visible artifact locations. Furthermore, we propose an algorithm for deriving the final low-pass filter-strength control signal and the integration of this signal with an existing edge-preserving low-pass filter.

6.4.1 Algorithm for spatial block-based artifact-location detection

This section presents our proposed algorithm for deriving locations with potentially visible coding artifact in the spatial domain (from now on indicated as artifact locations). The algorithm follows the functional system decomposition as depicted in Fig. 6.5.

A. Activity metric

In order to avoid detection of individual edges, the *activity* metric in the spatial domain is based on a block-based operation involving the computation of the

2D Sum of Absolute Differences (SAD), specified by

$$SAD = \sum_{y=1}^M \sum_{i=0}^{N-1} |p(x+i, y) - p(x+i+1, y)| + \sum_{x=1}^N \sum_{j=0}^{M-1} |p(x, y+j) - p(x, y+j+1)|. \quad (6.1)$$

Hereby, $p(x, y)$ denotes a start 2D pixel position at spatial block-grid location (x, y) , while M and N denote the width and height of the block. This metric is calculated for each block constructing the artifact-location detection kernel.

B. SAD classification

Each calculated SAD metric is classified employing discriminative features. Due to the nature of the applied SAD metric, the SAD value provides an indication for the amount of intra-block variation. Hereby, a low SAD value results in a “flat and/or low-frequency” region, while a medium SAD value indicates a “potentially artifact contaminated” region and finally a high SAD value corresponds to a “texture” region. The possible broad range of SAD values is partitioned into the previously indicated classification intervals by means of a set of thresholds, see Fig. 6.7(b) and Table 6.1. It should be noted that the thresholds refer to the cumulative SAD value, i.e. the threshold involves also intervals with a lower SAD value. For example, a potentially contaminated region may contain a “flat and/or low-frequency” background with a superimposed noise pattern leading to a medium SAD value. This leads to the consideration that a medium-valued SAD block is potentially contaminated by noise or intended medium texture. For simplicity, we hypothesize that a center block is filled with coding noise and verify our hypothesis by inspecting the spatial context of this center block. For each block, except block H, constructing the artifact detection-kernel, see Fig. 6.7(a), the calculated SAD for that block is classified into one of the three intervals “flat and/or low-frequency” region, “texture” region or none of these two cases leading to medium texture region, see Fig. 6.7(b). For the center block only, thus the SAD value of block H, Table 6.1 does not apply, as this block is only classified in “texture” or “non-texture” region. Since

Table 6.1 — *Threshold definition for spatial-domain SAD classification ($T_t > T_l$).*

Threshold	Region classification	SAD interval
T_l	flat and/or low-frequency	$SAD \leq T_l$
T_t	potentially artifact contaminated	$SAD < T_t$
T_t	texture	$SAD \geq T_t$

“non-texture” may mean that block H is noise contaminated, a further spatial analysis is then required involving the inspection of the surrounding blocks.

C. Artifact-location detection kernel

For each pixel in the image, located at the center of block H, an artifact-location detection kernel is analyzed using a set of surrounding blocks, which overlap with one border pixel, see Fig. 6.7(a). This kernel is shifted in a sliding window fashion across the pixels constructing the video lines. Basically, the chosen block size for the detection kernel in the spatial domain is defined to a fixed setting of either 3×3 or 5×5 pixels. This choice depends on the video resolution, which is either native (SD/HD) or upscaled (3×3 and 5×5 pixels, respectively). An odd block size avoids the detection to be biased in a certain direction. In order to detect small-sized flat regions, the block size of center block H must be kept small. Although we have employed square blocks, this is not mandatory. A non-square block may be chosen to either: (1) extend the detection area, or (2) influence the SAD sensitivity.

Potential artifact-location detection. Figure 6.7(c) depicts the full artifact-detection system operating in the spatial domain. At the left-hand side, 7 video

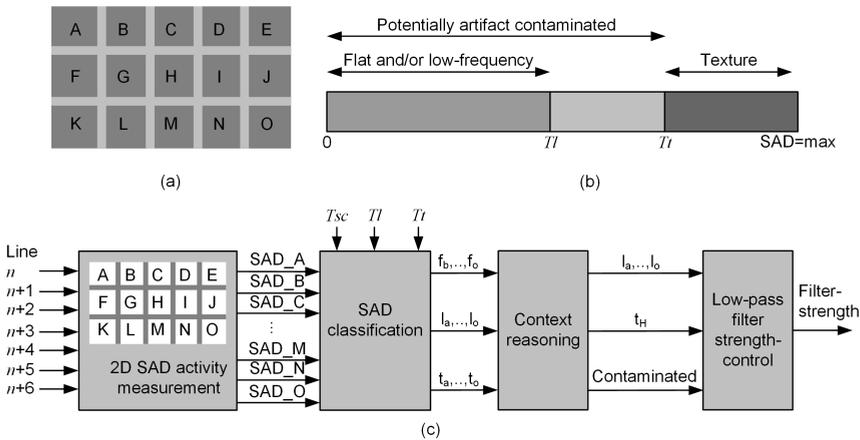


Figure 6.7 — Spatial-domain artifact-location detection system for visual artifacts based on fixed 3×3 blocks. (a) Construction of detection kernel deploying one border pixel of overlapping for the 3×3 blocks (light gray color). (b) Positioning of threshold in SAD range. (c) Basic block diagram of the spatial-domain artifact-detection system.

lines form the input for the detection kernel, which is constructed on the basis of 15 one-pixel overlapped blocks, with a fixed block size of 3×3 pixels, as an example. For each block in the detection kernel, the SAD-based activity metric is calculated. This results in 15 block-based SAD values (SAD_A,...,SAD_O), which are classified using two thresholds T_l and T_t . This classification results in two Boolean signals per block, leading to two Boolean sets $L = \{L_a, \dots, L_o\}$ and $T = \{T_a, \dots, T_o\}$, revealing the “flat and/or low-frequency” feature and “texture” feature. On the basis of context reasoning, exploring the presence of “flat and/or low-frequency” region and “texture” regions around the center block H, the decision concerning artifact contamination for block H is derived, leading to a Boolean signal “Contaminated”. This decision making is based on a set of rules, see Table 6.3, describing the possible noise occurrence cases that may arise from surrounding “texture” and “flat” block patterns. For the actual pixel location in the center of block H, a locally-adaptive control signal for the low-pass filter is derived.

Sensitivity control for perceptual tuning. The above design does not consider the noise-pattern visibility and the visual influence of the filter control. Therefore, a sensitivity control is added, which enables tuning of the detection performance. This tuning attempts to control the visibility of the artifact noise pattern, which is typically well visible in large “flat and/or low-frequency” regions. For control purposes, an additional threshold T_{sc} , see Fig. 6.7(c), is added to the artifact-location detection, which explicitly tests on the presence of significant coherent areas in the kernel related to “flat and/or low-frequency” regions. When the threshold T_{sc} is increased, the noise detection sensitivity becomes lower, leading to more blocks classified as “flat and/or low-frequency”. This results in a higher probability of potential noise contaminated regions and thus corresponding filtering.

Table 6.2 — *Threshold values for native and upscaled video using an artifact-location detection kernel with square and unequally-sized blocks and an 8-bit SAD value.*

Threshold	Thresholds native resolution			Threshold upscaled resolution
	kernel size			kernel size
	11×7	19×11	21×13	21×13
T_{sc}	0 - 255	0 - 255	0 - 255	0 - 255
T_l	0 - 150	0 - 150	0 - 150	0 - 80
T_t	150 - 255	200 - 255	200 - 255	160 - 255

Table 6.3 — Parameters and rules for artifact-location detection in the spatial domain, based on context reasoning.

Parameter	Nature	Rules
Classification results		
L	binary	$(l_B, l_C, l_D, l_G, l_I, l_L, l_M, l_N)$
T	binary	$(t_A, t_B, t_C, t_D, t_E, t_F, t_G, t_H, t_I, t_J, t_K, t_L, t_M, t_N, t_O)$
Spatial-domain “sensitivity” status information		
F	binary	$(f_A, f_B, f_C, f_D, f_E, f_F, f_J, f_K, f_L, f_M, f_N, f_O)$
F_{top}	binary	$(f_A \wedge f_B \wedge f_C \wedge f_D \wedge f_E)$
F_{bottom}	binary	$(f_K \wedge f_L \wedge f_M \wedge f_N \wedge f_O)$
F_{left}	binary	$(f_A \wedge f_F \wedge f_K)$
F_{right}	binary	$(f_E \wedge f_J \wedge f_O)$
F_{flat}	binary	$(F_{top} \vee F_{bottom} \vee F_{left} \vee F_{right})$
Spatial-domain “flat and/or low-frequency” status information		
L_{row1}	binary	$(l_B \vee l_C \vee l_D)$
L_{row2}	binary	$(l_G \vee l_I)$
L_{row3}	binary	$(l_L \vee l_M \vee l_N)$
Low	binary	$(L_{row1} \vee L_{row2} \vee L_{row3})$
Spatial-domain “texture” status information		
T_{row1}	binary	$(t_B \vee t_C \vee t_D)$
T_{row2}	binary	$(t_G \vee t_I)$
T_{row3}	binary	$(t_L \vee t_M \vee t_N)$
Tex	binary	$(T_{row1} \vee T_{row2} \vee T_{row3})$
Spatial-domain “artifact contaminated” center-block status information		
$Block_Hcenter$	binary	$\neg t_H$
Center pixel mosquito noise / ringing contaminated		
$Contaminated$	binary	$(Low \wedge F_{flat} \wedge Tex \wedge Block_Hcenter)$

Empirical threshold settings. Table 6.2 indicates typical threshold values for native and upscaled video and an activity detection kernel with square blocks and rectangular blocks.

D. Spatial noise-pattern analysis

In order to determine artifact contamination for block H, context reasoning is conducted involving a set of Boolean equations. These equations describe the properties of the SAD value distribution over the detection kernel. The satisfaction of each SAD value to the imposed thresholds results in a binary indicator, being either *true* or *false*. In order to encompass the sensitivity control with its threshold T_{sc} , an additional Boolean set F is introduced, indicating

whether a block is potentially part of a flat area in the detection kernel. Hence, the evaluation results in three sets L , F and T , respectively, referring to “flat and/or low-frequency”, “part of flat area” and “texture”. Each set contains the binary classification values for the blocks constructing the detection kernel. The Boolean equations describing the context reasoning for potential artifact detection are presented in Table 6.3. The equations describe the context reasoning process associated to the right-hand block in Fig. 6.7(c). Although the processing is block-based, the result of this reasoning process is a pixel-based decision function, indicating whether the center pixel of block H is artifact contaminated.

Let us start with the middle of the presented Table 6.3. The surrounding blocks of center block H are tested to have at least one block of “flat and/or low-frequency” and at least one block of “texture”. If so, this defines the value of the “Low” and “Tex” indicators to be true. Furthermore, if center block H is not textured, see the bottom of the table, artifact contamination is considered possible. Finally, the possible visibility of the coding noise is evaluated with the flat area test. This test involves the verification that the complete row of top or bottom blocks of the kernel are flat, or the two columns at the left and right of block H are flat. The binary indicator “Flat” becomes true if one the four conditions occurs. The final filtering switches on for the situation that all four evaluations are true.

6.4.2 Algorithm for frequency-domain artifact-location detection

This section presents our proposed algorithm for deriving artifact locations in the frequency domain, as an alternative to the spatial-domain based solution of the previous section. The algorithm follows the same functional diagram of the system decomposition, as depicted in Fig. 6.5.

A. Adaptations resulting from employing the frequency domain

Although the frequency-domain solution has a high resemblance with the already proposed spatial-domain concept, it differs with respect to: (1) the involved activity metric which is now related to the frequency domain, (2) quantization of the calculated energy or a simplified form of it, and (3) specific classification of the activity associated with the region characterization. It will become clear that analysis in the frequency domain enables a more refined characterization, since artifacts such as ringing, can be well identified. The frequency transformation is not based on an FFT, but employs a 4×4 integer DCT. The use of the DCT is beneficial for two purposes: (1) the transform offers a real and fast implementation suited for embedded usage, (2) the DCT is already required for the decoding of DTV signals. The DTV decoding is based on 8×8 DCT blocks, in which a 4×4 DCT can be included easily. In the above paragraph, the term

frequency is regularly used. The DCT produces spatial frequencies (u, v) after transformation, which are employed to determine noise patterns and associate critical locations. In the remaining part of this chapter, we denote the usage of the spatial frequencies (u, v) as detection in the frequency domain, so that notation is simplified. Signal processing in the frequency domain offers a more refined analysis of the noise patterns resulting from video compression. Compared to the spatial domain, we are now able to extend the analysis to measure specific noise patterns such as “mosquito noise” and “ringing”. For this reason, we present a set of four typical noise patterns, which potentially occur in DTV-coded video content. These noise patterns are shown in Fig. 6.8. Compared to the spatial domain, we have added two patterns specifically representing coding noise: (1) ringing and (2) mosquito noise. The measurement of these patterns can be efficiently implemented in the frequency domain by analyzing the occurring AC coefficient patterns resulting after DCT transformation. Ringing patterns as indicated in the figure, can be identified by applying energy-based ranking in combination with accumulating the coefficient energy of the non-pure horizontal (or vertical) AC coefficients. These pure directional structures are covered by the first row (or column) of the obtained coefficient matrix. The energy ranking mechanism avoids the detection of individual low-

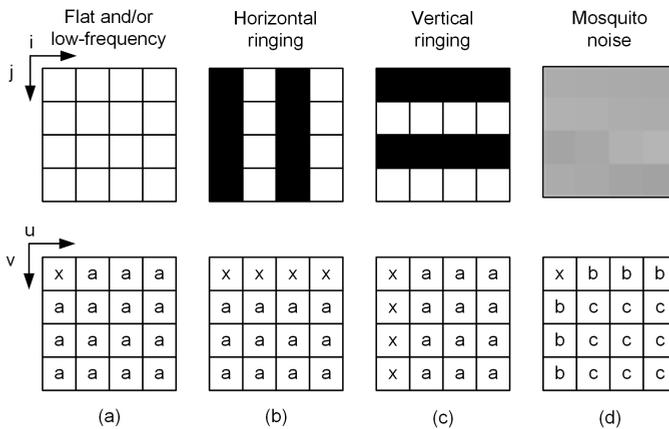


Figure 6.8 — Video coding artifacts used for activity measurement in the transform domain and the corresponding DCT coefficient subsets. (a) “Flat and/or low-frequency” region. (b) “Horizontal ringing” region. (c) “Vertical ringing” region. (d) “Mosquito noise” contaminated. Only coefficients with label *a* are used for energy calculation in subfigure (a), (b) and (c). For subfigure (d), the coefficients with label *b* and *c* are used for magnitude comparison.

or high-frequency patterns, thereby simplifying detection of the intended noise patterns. As in the spatial-domain case, the coefficient patterns are not unique and involve the surrounding blocks covering the larger spatial context to provide information whether the noise pattern is a single occasional event or part of a larger structure. This remark also holds for the detection of mosquito noise.

B. Block-based pre-processing

The activity metric in the transform domain can be defined, using a 2D DCT according to $Y = AXA^T$. Matrix $X(i, j)$ has a size of 4×4 samples and A denotes a 4×4 integer DCT transform matrix, where A is applied for horizontal and vertical transformation (separability of the 2D DCT). When extending this transformation with a quantization stage Q_f , the final equation becomes $Y_Q = AXA^T \otimes Q_f$. Hereby, Q_f involves a post-scaling operation conducted at each individual DCT coefficient, implementing a scalar quantizer identical to the standard version as employed in H.264 coding [40]. In this way, a total of 52 weighted quantization values, in the range of 0-51, are available for compression. After transformation and quantization, the resulting quantized coefficient matrix Y_Q has DCT domain coordinates (u, v) and can be used to compute the DCT coefficient energy or a related version of it, on the basis of the obtained quantized set of 4×4 DCT coefficients. The coefficient with $(u, v) = (0, 0)$ is called the DC coefficient (average block value) and all other coefficients are defined as AC coefficients ($(u, v) \neq (0, 0)$). An important design parameter is the choice of the adopted block size for DCT transformation. In the frequency-based system, we want to exploit a comparable block size for artifact detection and associated processing, as introduced in the spatial-domain algorithm, leading to a final block size of 4×4 pixels.

Table 6.4 — *Threshold definition for frequency-domain energy classification.*

Threshold	Region classification	Energy interval
T_{Fl}	flat and/or low-frequency	$E_{Fl} \leq T_{Fl}$
T_{Hedge}	Horizontal ringing	$E_{Hedge} \leq T_{Hedge}$
T_{Vedge}	Vertical ringing	$E_{Vedge} \leq T_{Vedge}$
T_b	Mosquito noise	$ Y_Q(u, v) < T_b$ $(u = 0 \vee v = 0) \wedge ((u, v) \neq (0, 0))$
T_c		$ Y_Q(u, v) < T_c$ $(u > 0 \wedge v > 0)$

C. Energy pattern measurements and classification

Due to the broad range of energy distributions, i.e. the absence and variations of AC coefficients at particular (u, v) locations and the associated AC coefficient amplitudes, we propose a generic calculation approach in combination with an energy ranking method to conduct the final classification.

Energy and comparison metric. In order to be sufficiently sensitive for low- or high-frequency patterns, the AC coefficient energy is a suitable metric, as it involves the accumulation of squared coefficients. The AC energy calculation is performed according to $E_{AC} = \sum_{AC} Y_Q(u, v)^2$. This metric generates a broad dynamic range of energies even for small-sized blocks, enabling robust detection of specific noise patterns, see Fig. 6.8. However, there is one exception on this calculation concept and that is the detection of mosquito noise. This pattern is appearing as a random noise pattern with low/medium AC coefficient amplitudes for which the squaring operation does not aid in the detection. This particularly holds for distinguishing mosquito noise patterns from intended texture. This is a rather fundamental problem that cannot be uniquely solved in an easy way. We have observed a large set of mosquito-noise contaminated images and studied the AC coefficient appearance in detail. From the empirical evaluation of the mosquito noise appearances, we have concluded that the frequency components constructing the noise pattern have low/moderate amplitudes, while the majority of frequency components are non-zero. For this reason, we make a reasonable assumption about the noise and the way for separating it from texture. Instead of squaring the coefficients and computing the energy, we apply an amplitude test that matches with the previous observation about coefficient magnitudes. This amplitude test involves a simple comparison with a threshold. Also for mosquito noise, the noise assumption is then further validated by employing the concept of context reasoning using the surrounding block data patterns, as used in the previous section.

Energy ranking. Let us now discuss how the actual ranking of the detection of patterns is organized. The calculated energies for the four possible coding artifacts of each DCT block are classified on the basis of a set of thresholds, where each coding artifact has its own energy threshold, see Table 6.4. The key in the final classification is that the coding artifacts are ranked to their importance. First, if the most important artifact is not measured, then the next priority artifact is tested on its presence, etc. Second, if one of the artifact measurements satisfies its threshold condition, then the DCT block is assigned the corresponding coding artifact classification. Third, for the situation that none of the supported artifacts is present, the DCT block is classified as “texture” (without visible noise), see Fig. 6.9, which is also the default setting (see Table 6.5). Let us now implement this approach.

Noise-pattern classification. Prior to elaborating on the ranking process, we define a set X , which contains the elements x_i , with i being a block index referring to the surrounding blocks $\{A,B,C,\dots,M,N,O\}$, constructing the detection kernel. For each block, the ranking process results in a final region classification, which is stored in x_i . The energy calculation and the associated ranking is depicted in Table 6.5. At the top of this table (Step 0), block x_i is initialized and obtains the classification value “texture” and can be modified by one of the four coding artifacts, while sequentially processing the rows constructing Table 6.5.

Let us now further detail the ranking process as depicted in Table 6.5. The most important noise visibility occurs in “flat and/or low-frequency” regions. Therefore, we start with detecting this type of regions (Step 1). To this end, we measure the AC energy of the whole DCT block. If this energy is exceeding the associated threshold, then there is chance that this noise occurs due to ringing or similar patterns. Therefore, we refine the detection of possible noise with respect to ringing. We do this by measuring the energy associated with the non-ringing parts of the AC coefficients. This non-ringing part is the AC coefficient block without the first column or row of coefficients. This non-ringing part is analyzed with respect to the AC energy and if this is sufficiently low, the energy is contained in the first row and/or column (Step 2 and 3). This coefficient pattern may involve ringing, which can be validated by context reasoning of the surrounding blocks. If the AC energy of the non-ringing part is higher than the associated threshold, the DCT block is tested for mosquito noise (Step 4), as discussed above. If none of the four noise patterns apply, the block remains texture classified. For the first three coding noise patterns, the AC coefficients are squared, whereas for the fourth coding noise pattern called “mosquito noise”, the classification compares coefficient magnitudes where all “ b ” coefficients must be smaller than their threshold T_b and all “ c ” coefficients smaller than their threshold T_c . The overview of these individual tests is given in Table 6.5 at the bottom. Hereby, the second top row in the table indicates the always computed energy value based on the AC coefficients for the three noise types evaluated in Steps 1, 2 and 3. The last row indicates the AC-coefficient magnitude comparison for mosquito noise detection (Step 4).

Example of ringing detection. The transformation of a 4×4 region contaminated with vertical ringing results in an AC coefficient matrix Y_Q , as depicted in the lower part of Fig. 6.8(c). The energy of interest is located at the “ \times ” locations of the coefficient matrix Y_Q in that figure, while theoretically the energy at the “ a ” locations is zero. When calculating the energy to verify the “flat and/or low-frequency” region, all AC coefficients are squared and accumulated. Due to presence of vertical ringing, the final calculated energy will not satisfy the energy comparison associated with the “flat and/or low-frequency” region. The

Table 6.5 — Block-based energy calculation for frequency-domain feature classification. Input to the classification is formed by the set of AC coefficients, $Y_Q(u, v)$ for $(u, v) \neq (0, 0)$.

Parameter	Specification
Step 0: Block feature initialization	
x_i	= Tex
Equation for flat and/or low frequency and ringing detection	
E_x	$= \sum_u \sum_v Y_Q(u, v)^2$.
Step 1: Flat and/or low-frequency detection ($x = Fl$)	
E_{Fl} x_i	where $u \in \{0, 1, 2, 3\}, v \in \{0, 1, 2, 3\}, (u, v) \neq (0, 0)$ if $E_{FL} < T_{FL}$ then Fl
Step 2: Vertical ringing detection ($x = Vedge$)	
E_{Vedge} x_i	where $u \in \{0, 1, 2, 3\}, v \in \{1, 2, 3\}$ if $E_{Vedge} < T_{Vedge}$ then Vedge
Step 3: Horizontal ringing detection ($x = Hedge$)	
E_{Hedge} x_i	where $u \in \{1, 2, 3\}, v \in \{0, 1, 2, 3\}$ if $E_{Hedge} < T_{Hedge}$ then Hedge
Step 4: Equation for mosquito noise detection	
E_{Mos} x_i	$\begin{cases} 1, & \text{if } (u = 0 \vee v = 0) \wedge ((u, v) \neq (0, 0)) \\ & \forall Y_Q(u, v) = b < T_b \wedge \\ & (u > 0 \wedge v > 0) \forall Y_Q(u, v) = c < T_c, \\ 0, & \text{otherwise.} \end{cases}$ if $E_{Mos} == 1$ then Mos

energy calculation associated with the vertical ringing pattern excludes the AC coefficients located at the “ \times ” positions, as these AC coefficient values can vary from almost zero up to 9-bit values for 8-bit video data. However, the AC coefficients located at the a locations are included in the energy calculation, as these coefficients should be theoretically zero. When the sum of squared coefficient energy terms from the “ a ” locations satisfy the vertical energy threshold condition, the 4×4 region is classified as vertical ringing (Vedge is set), according to the ranking depicted in Table 6.5.

D. Artifact-location detection kernel

For the final detection of potential noise patterns in center block H, we hypothesize that a center block is filled with coding noise and verify our hypothesis by inspecting the spatial context (the surrounding blocks) of this center block. For each region covered by the center block H, a detection kernel is analyzed

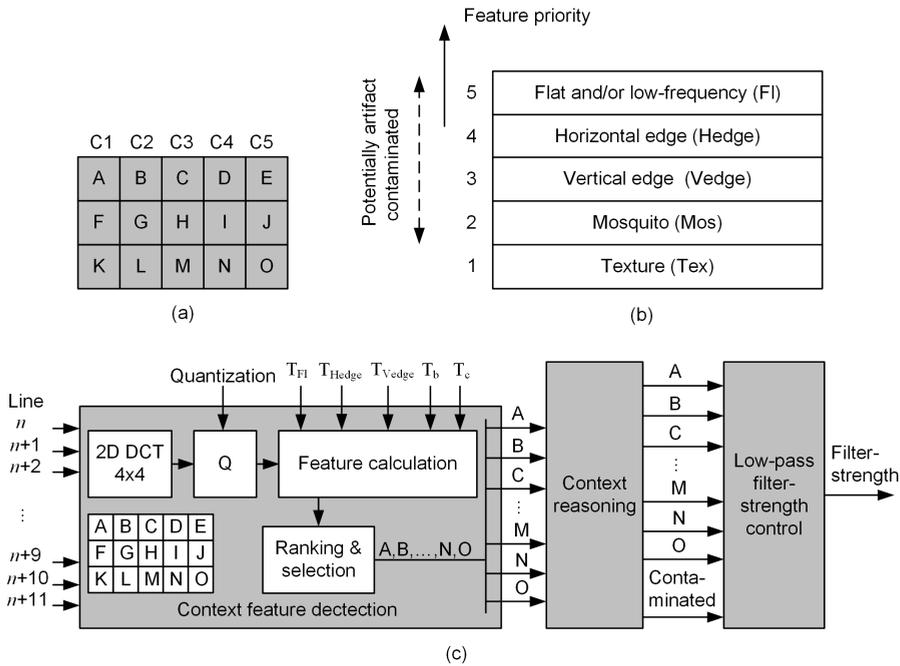


Figure 6.9 — *Frequency-domain artifact detection. (a) Transform-block locations. (b) Feature ranking, lowest number refers to highest priority. (c) Basic block diagram of the frequency-domain artifact-detection system.*

using a set of non-overlapping surrounding 4×4 pixel blocks, see Fig. 6.9(a). This kernel is shifted in a sliding-window fashion across the image on a block-by-block basis.

Potential artifact-location detection. Figure 6.9(c) depicts the basic block diagram of the artifact-detection system operating in the frequency domain. At the left-hand side, three block stripes (12 video lines) form the input for the window-based detection kernel. For each block in the detection kernel, the DCT-based activity metric is calculated. Prior to energy classification, the energy may be quantized, thereby locally smoothing the image in a 4×4 pixel block. The energy classification involves four conditions, which are depicted in Table 6.5. After energy ranking, see Table 6.5, each block $x_i \in \{A, B, \dots, N, O\}$ is assigned its final feature. On the basis of context reasoning, using this set of block-oriented features, the potential contamination with “mosquito noise” or “ringing” in block H is derived, leading to a Boolean signal “contaminated”.

The derivation of this binary signal involves a similar process compared to the spatial domain, for detecting “mosquito noise” contamination from the previous section. However, a separate detection is conducted to reveal the presence of potential “ringing” in block H. The detection of potential “ringing” involves a direct relation with a “texture” region and its visibility enabled by a “flat and/or low-frequency” region. The upper part of Table 6.7 show the rules for detecting “mosquito noise”, while the lower part depicts the rules for “ringing” detection in the frequency domain. The final calculation of the Boolean signal “contaminated” is depicted at the bottom of Table 6.7.

Empirical threshold settings. Table 6.6 indicates typical threshold values for noise-pattern detection in the frequency domain involving native and upscaled video. For all rows in the table with range 0– n , the corresponding quantization value Q_f is in the range of 30 – 4, respectively.

E. Frequency artifact noise-pattern analysis

In order to determine artifact contamination for block H, context reasoning is conducted with a set of Boolean equations, involving the detection kernel block features x_i . These equations describe the properties of the block-based feature distribution over the detection kernel. The final artifact detection is based on four aspects, each captured by a final Boolean variable. The Boolean equations describing the context reasoning for potential artifact detection are presented in Table 6.7. The equations describe the context-reasoning process performed by the right-hand block in Fig. 6.9(c). As the processing is block-based, the result of this reasoning process is a block-based decision function, indicating whether

Table 6.6 — *Thresholds for 8-bit native and upscaled video using an activity-detection kernel in the frequency domain with 10-bit AC coefficients and 24-bit energy values. For each row, the quantization value Q_f is in the range of 30 – 4.*

Threshold	Value native resolution kernel size 20 × 12	Value upscaled resolution kernel size 20 × 20
T_{Fl}	0 - 60	0 - 40
T_{Vedge}	0 - 10	0 - 10
T_{Hedge}	0 - 10	0 - 10
T_b	0 - 40	0 - 30
T_c	0 - 30	0 - 20

Table 6.7 — Parameters and rules for spatial/context reasoning, based on surrounding blocks for final coding noise detection in the frequency domain.

Parameter	Reasoning rules
Frequency-domain texture detection	
T_{xtr}	$\begin{cases} 1, & \exists x_i \in X, x_i \neq \{A, F, K, H, E, J, O\} \text{ where } x_i = Tex; \\ 0, & \text{otherwise.} \end{cases}$
Frequency-domain flat and/or low-frequency detection	
F_{c1A}	$\begin{cases} 1, & ((x_B = Fl \vee x_G = Fl \vee x_F = Fl) \wedge x_A = Fl); \\ 0, & \text{otherwise.} \end{cases}$
F_{c1F}	$\begin{cases} 1, & ((x_B = Fl \vee x_G = Fl \vee x_L = Fl) \wedge x_F = Fl); \\ 0, & \text{otherwise.} \end{cases}$
F_{c1K}	$\begin{cases} 1, & ((x_G = Fl \vee x_L = Fl \vee x_F = Fl) \wedge (x_K = Fl)); \\ 0, & \text{otherwise.} \end{cases}$
F_{c1}	$F_{c1A} \vee F_{c1F} \vee F_{c1K}$
F_{lat}	$F_{c1} \vee F_{c2} \vee F_{c3} \vee F_{c4}$
Frequency-domain center block contamination	
C_{ntr}	$\begin{cases} 1, & (x_H = Fl) \vee (x_H = Mos); \\ 0, & \text{otherwise.} \end{cases}$
Frequency-domain detector ringing detection	
R_b	$\begin{cases} 1, & (x_M = Fl \wedge x_H = Vedge \wedge x_C = Tex); \\ 0, & \text{otherwise.} \end{cases}$
R_l	$\begin{cases} 1, & (x_I = Tex \wedge x_H = Hedge \wedge x_G = Fl); \\ 0, & \text{otherwise.} \end{cases}$
R_r	$\begin{cases} 1, & (x_G = Fl \wedge x_H = Hedge \wedge x_I = Tex); \\ 0, & \text{otherwise.} \end{cases}$
R_t	$\begin{cases} 1, & (x_C = Fl \wedge x_H = Vedge \wedge x_M = Tex); \\ 0, & \text{otherwise.} \end{cases}$
R_{ing}	$R_b \vee R_l \vee R_r \vee R_t$
Frequency-domain center block artifact contaminated	
$Contaminated$	$(F_{lat} \wedge T_{xtr} \wedge C_{ntr}) \vee R_{ing}$.

block H is contaminated with “mosquito noise” or “ringing”.

Let us start at the top of the presented Table 6.7. Block H is “mosquito noise” contaminated when: (1) the surrounding blocks of center block H at least have one block of “texture” and at least two neighboring blocks of “flat and/or low-frequency”. This “flat and/or low-frequency” region is determined for four columns $C1, C2, C3, C4$, see Fig. 6.9(a), using neighboring block relations $F_{c1A}, F_{c1F}, F_{c1K}$, as depicted in Table 6.7. (2) Block H is either “flat and/or

low-frequency” or “mosquito noise”, as depicted in the middle of Table 6.3. Block H is “ringing” contaminated when: (1) block H is either “horizontal ringing” or “vertical ringing” and (2) this block is preceded by a “flat and/or low-frequency block” and succeeded by a “texture” block or vice versa, as depicted in the lower part of Table 6.3. The results of the previous individual conditions are combined in a final Boolean equation, as depicted at the Bottom of Table 6.7. The final filtering switches on for the situation that the mosquito noise or ringing detection is true.

6.4.3 Algorithm for control of the filter strength by entropy-based low-pass filtering

On the basis of the artifact-location signal, a filter-strength control signal with a diamond-shaped aperture is derived, avoiding the introduction of new artifacts, resulting from the switching behavior of the low-pass filter. The applied low-pass filter has been adopted from [129], because of its flexibility to adapt the actual filter aperture and the preservation of strong edges. The filter-strength signal controls the low-pass filter in three ways: (1) it indicates the pixel locations that need to be smoothed, (2) it chooses the applied filter aperture to be either 3×3 or 5×5 (more smoothing), and (3) it modifies the filter coefficients according to [129]. The derivation of the final filter-strength con-

Algorithm 15 2D expansion of spatial artifact-location detection signal

Require: Detection kernel block features

Ensure: Smooth filter strength decline

Set maximum filter strength s_c for detected artifact-location

▷ Adaptive filter strength expansion in top, left

▷ bottom, right, anti-diagonal and diagonal direction

for $i = direction$ **do**

if $direction == flat$ **then** ▷ Check flatness in direction

 Maximal aperture

$\alpha = 1$ ▷ Set decline rate

else

 Minimal aperture

$\alpha = 2$ ▷ Set decline rate

end if

end for

At position $(i - n, j - m)$ $tmp = \max(s_c - \alpha |n - |m||, 0)$

$g(i - n, j - m) = \max(g(i - n, j - m), tmp)$

for $-N \leq m \leq +N$ **and** $-N - |m| \leq n \leq +N - |m|$

control signal is based on an adaptive 2D expansion of the artifact-location signal, which is elaborated hereafter.

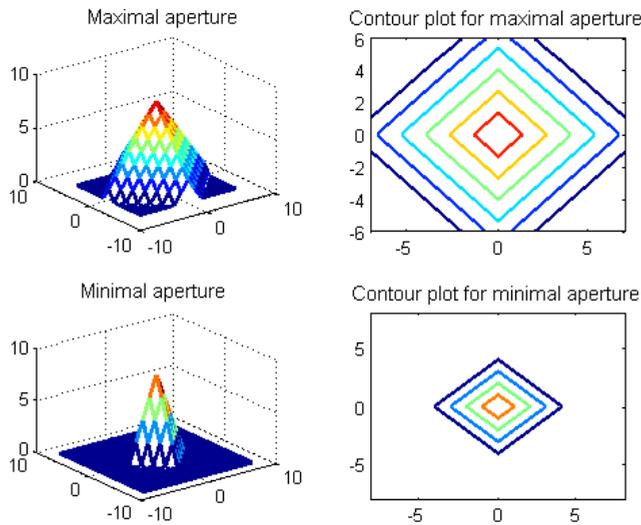


Figure 6.10 — *Aperture of the filter-strength control signal for spatial-domain detection. All parameters along the axes are relative distances in pixels counted from pyramid center.*

A. Spatial-domain filter-strength control signal and expansion

For the spatial-domain detection system, the aperture of the artifact-location signal depends on the deployed block size constructing the detection kernel and the presence of “flat and/or low-frequency” regions. The minimal and maximal apertures are depicted in Fig. 6.10. The plots are given for clarity, because in practice the strength value of the control signal will not be so straight and the strength distribution will not be equal in all directions simultaneously (at least in one direction, there should be texture to create visible artifacts). In these artificial plots, the maximal aperture is shown with a detection kernel based on 5×5 pixel blocks, while the minimum aperture is shown for a detection kernel of 3×3 pixel blocks. The diamond-shaped aperture is calculated according to Algorithm 15. The specified calculation involves a diamond-shaped coefficient-pattern construction that is symmetric in both horizontal and vertical directions. The filter-strength setting is restricted to two values only, thus

$\alpha = \{1, 2\}$. For $\alpha = 2$, the control signal decline is faster, so that the effective aperture is halved.

The processing is conducted on a pixel-by-pixel and line-by-line basis. For each individual pixel, the artifact-detection signal leads to a diamond-shaped control signal overlapping the surrounding pixels. The final strength of the control signal of a pixel is determined when all individual pixels have been addressed. This means in practice that a previously calculated control signal for a pixel can be re-assigned to another value. In order to avoid that a particular pixel obtains a lower control value from a successive calculation, the largest filter-strength control value is maintained by incorporating a *max* function in the algorithm.

In the presented algorithm, there are two *max* functions applied for the following reasons. The first *max* function prevents negative values outside the aperture and bounds the strength signal to zero. The second *max* function avoids a pixel location to be assigned a filter-strength signal that is lower compared to a previously calculated strength value. Such a situation may happen when the pyramidal pattern of Fig. 6.11 is shifted from the current pixel to the next pixel, or from the current line to the next line. When shifting this pyramidal pattern, while applying the *max* functions, the final control signal pattern is effectively expanded over a larger area. This explains the term 2D expansion of the effective filter-control signal.

B. Frequency-domain filter-strength control signal

The derivation of the filter-strength control signal for the frequency-domain

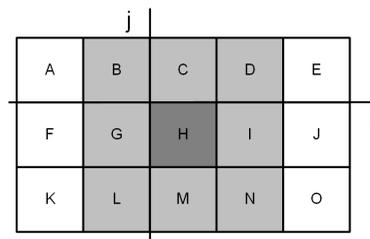


Figure 6.11 — Relation between the artifact-detection kernel and the filter-strength aperture due to 2D expansion. Here, the dark center block indicates the potential noise-contaminated block with the highest filter-strength value. The gray-valued blocks indicate the decreasing filter-strength region and the white blocks have a zero-valued filter strength.

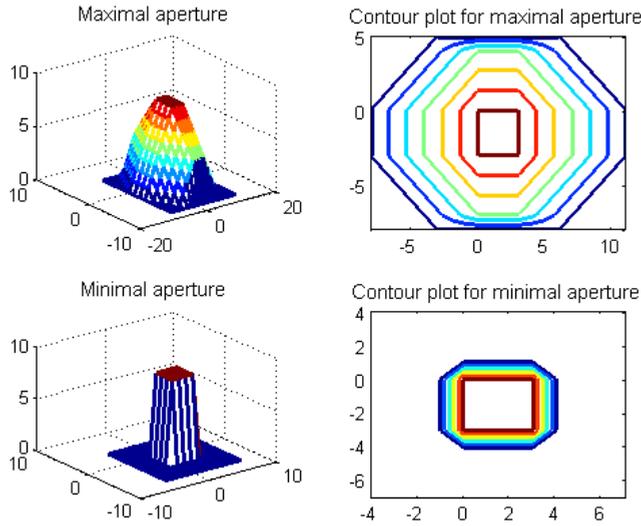


Figure 6.12 — Aperture of the low-pass filter-strength control signal for frequency-domain detection. All parameters along the axes are relative distances in pixels counted from the upper-left corner of the center block.

detection system differs from the spatial domain in the sense that the artifact-location detection signal is now block-based, resulting in a block-based filter-strength control signal. For the situation that center block H is contaminated with coding noise, all pixels constructing block H are filtered, see Fig. 6.11. As a result, the filter-strength control signal is “hat”-shaped, see Fig. 6.12. The filter-strength decline is obtained in a similar manner compared to the spatial domain and starts at the block boundary. The plots are again based on an illustrative example for clarity. The calculation, including the two *max* functions, is performed according to Algorithm 16, which follows a similar approach as discussed for the spatial domain, while the actual filter-strength signal is calculated on a per block basis.

C. Control of the entropy-based low-pass filter

The generated filter-strength control signal is employed by the entropy-based low-pass filter depicted in Fig. 6.13. The calculated filter-strength control signal from the previous subsections A and B is supplied to the adaptive low-pass

filter (at the top of the figure). The complete filter of [129] also involves an entropy calculation and an edge detection. The filter-strength signal computed by our system is translated via a Look-Up Table (LUT) into an entropy value, thereby effectively controlling the filtering. Moreover, the control signal reveals the pixel locations where filtering should be applied. The edge detection is based on computing the local dynamic range, so that pixels located nearby an edge are protected against low-pass filtering. This avoids strong edges from being blurred. This mechanism overrules our computed filter-strength control signal to preserve sharpness.

6.5 Experiments and validation results of block-based artifact-detection

In this section, we present experimental results for the two proposed approaches of the artifact-location detection system and the resulting picture quality after filtering when such detection systems are applied. The first part of this section addresses the performance and scores of each detector at pixel level, where for each detector both visual and numerical results are presented. The second part of this section provides the results on filtering of the detected artifacts, when the previous detection systems are applied.

The used systems for detection experiments are according to the system descriptions presented in Section 6.4.1 and Section 6.4.2, in particular Fig. 6.7 and Fig. 6.9. The system used for adaptive low-pass filtering is presented in Section 6.4.3, according to Fig. 6.13. The derived filter-strength control signal for both approaches is supplied to an edge-preserving local-entropy-based low-pass filter [129], which is then actively filtering the coding noise.

All results have been obtained on the basis of a standalone software-based artifact-reduction simulation, employing both spatial-domain and frequency-

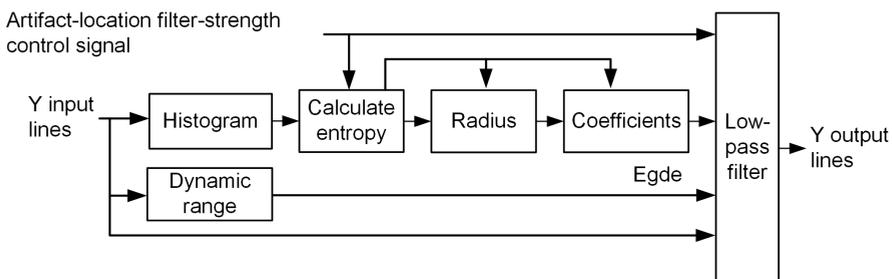


Figure 6.13 — Basic block-diagram of entropy-based local-adaptive LP filter.

domain detection processing. The spatial-domain-based detection system has been validated using an artifact-location detection kernel utilizing various block sizes, whereas the frequency-domain-based visual-artifact location detection system is validated for a fixed block size. Both approaches have been tested

Algorithm 16 2D expansion of frequency artifact-location detection signal

Require: Detection kernel block features

Ensure: Smooth filter strength decline

Set maximum filter strength for detected artifact-location

$g_H(i, j) = s_c$ $0 \leq i \leq B - 1$ and $0 \leq j \leq B - 1$ ▷ B is block dimension

▷ Adaptive filter strength expansion in top, left

▷ bottom, right, anti-diagonal and diagonal direction

for $i = direction$ **do**

if direction == flat **then** ▷ Check flatness in direction

$\alpha = 1$ ▷ Maximal aperture

else

$\alpha = 2$ ▷ Minimal aperture

end if

end for

 ▷ Calculate the strengths for all surrounding blocks

Choose block coordinates

for all pixels $(i - n, j - m)$ inside a block **do**

$top_diag = \max(s_c - \alpha m - \alpha n, 0)$ ▷ Top-left block

$top = \max(s_c - \alpha m, 0)$, ▷ Top-middle block

$top_antidiag = \max(s_c - \alpha m - \alpha n, 0)$ ▷ Top-right block

$left = \max(s_c - \alpha n, 0)$ ▷ Left-hand block

 etc.

 ▷ Final filter-strength assignment

$g_B(i - n, j - m) = \max(g_B(i - n, j - m), top_diag)$ ▷ Top-left block

$g_C(i + n - 1, j - m) = \max(g_C(i + n - 1, j - m), top)$ ▷ Top-middle block

$g_D(i + B - 1 + n, j - m) = \max(g_D(i + B - 1 + n, j - m), top_antidiag)$ ▷

Top-right block

$g_G(i - n, j + m - 1) = \max(g_G(i - n, j + m - 1), left)$ ▷ Left-hand block

 etc.

end for

on native video, while upscaled video results are in an appendix.

The chosen dataset with test images has been adopted from [123], which enables a fair comparison with results obtained from literature published in the same time period.

6.5.1 Evaluation of the detection approaches

We compare the artifact-detection performances of both spatial-domain and frequency-domain processing. For the spatial-domain detection, we have used a block size of 3×3 and 5×5 pixels, forming a detection kernel size of 11×7 and 21×13 pixels, respectively. Furthermore, we have used a detection kernel with an aperture of 19×11 pixels constructed from a combination of three different block sizes in one grid, where the center block has 3×3 pixels and its horizontal and vertical neighbors have rectangular block sizes, either 3×5 or 5×3 pixels. In this mixed grid, all remaining blocks have 5×5 pixels. The motivation for this mixed grid experiment is based on obtaining a better discrimination between intended texture and noise patterns, while preserving the detection of fine detailed edges. The frequency detection system is verified for a fixed block size of 4×4 pixels, resulting in a detection kernel aperture of 20×12 pixels. The motivation for this fixed size is found in the desire to align the detection block size with the 8×8 -pixel block grid employed for MPEG video coding in the DTV receiver.

A. Visual performance impression of the artifact-detection systems

Prior to detailing the experimental artifact-location detection results for spatial- and frequency-based detection, we first visualize for both domains, the detected pixel results for a representative image. This visualization gives a quick impression about the operation of the detection system.

Artifact-location detection in the spatial domain. Figure 6.14(b) shows that the 2D SAD value is suitable for clearly distinguishing flat regions from textured regions. However, note that the top face region of the face has low SAD values, although filtering should be prevented at such locations. This explains why the proposed algorithm requires that relatively large flat regions should be present for detection. Figure 6.14(c) reveals the detected pixels for possible filtering. The border of the hat is consistently detected, but the top face region contains several false-positive detections. In order to reduce the amount of false positives, the sensitivity of the detector is decreased. Figure 6.14(d) clearly shows the absence of these false detections due to a lower sensitivity

setting. However, this benefit is achieved at the expense of introducing small gaps between the detected regions around the object and a somewhat larger open space between the detected region and the object border. As discussed earlier, these gaps will be bridged later by employing a diamond-shaped expansion.

Artifact-location detection in the frequency domain. Figure 6.15(b) depicts the raw detected video patterns, without any context reasoning, for the situa-



Figure 6.14 — Visualization of artifact-location detection in the spatial domain. (a) Original image. (b) On a per-pixel-basis, the 2D SAD value in gray level, using 5×5 pixel blocks in the detection kernel. (c) In red color, the detected artifact pixels with maximal sensitivity. (d) Artifact-location detection with a lower sensitivity.

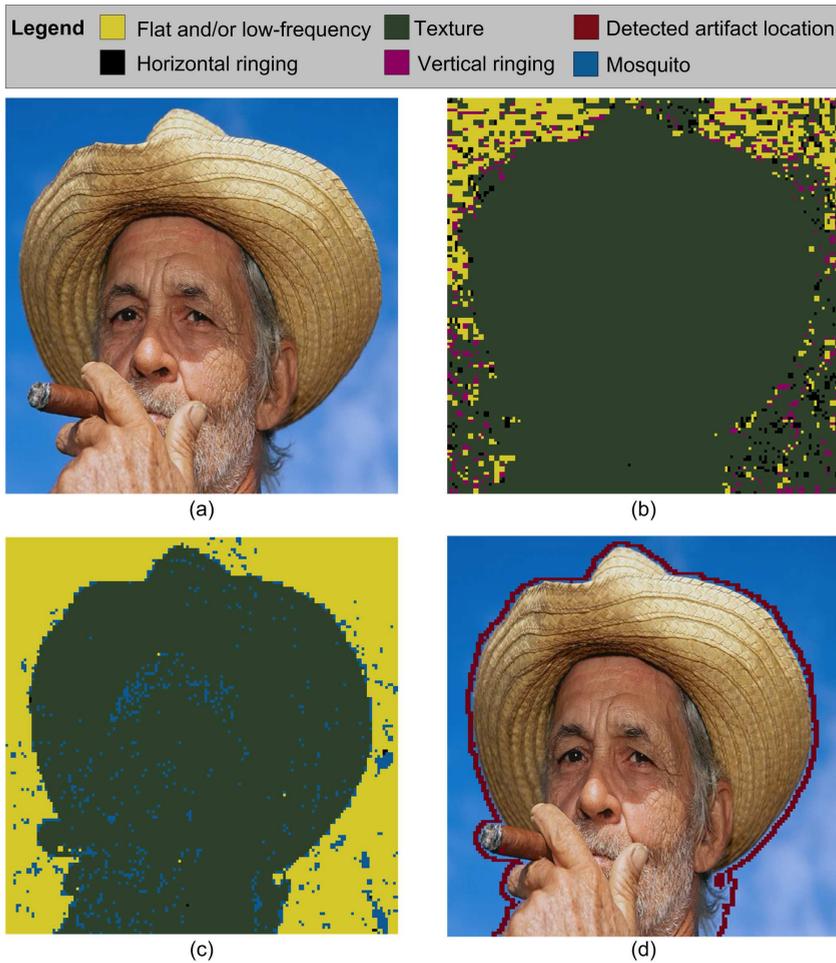


Figure 6.15 — Visualization of artifact-location detection in the frequency domain. (a) Original image. (b) Raw AC DCT-coefficient pattern classification where thresholds and quantization are switched off. (c) Raw AC DCT-coefficient pattern detection with thresholds set to non-zero and a quantization step size of 4. (d) Finally detected potential artifact locations.

tion that the detector thresholds (see Table 6.6) are set to zero and no coefficient quantization is applied. From this figure it becomes clear that without proper thresholding and coefficient quantization, the patterns have a more random behavior, due to the presence of low-amplitude high-frequency AC coefficients. We have applied coefficient quantization using a scalar coefficient quantizer as

used in the H.264 standard [40]. This quantization harmonizes the existing coefficient amplitudes, so that the image with detection becomes more consistent and thus suitable for context reasoning, see Fig. 6.15(c). Due to quantization, frequency ringing patterns virtually disappear and the flat region in the background becomes coherent in the detection. The remaining blue-colored blocks (see Fig. 6.15(c)) indicating potential mosquito noise, are scattered in the facial region and around the object. The proximity of flat regions in the background and the presence of texture surrounding the inner contour of the object lead then to positive artifact detections around the object contour, see Fig. 6.15(d).

B. Visible detection performance of the coding noise

Let us now detail both the subjective and the objective detection performance of potential artifact-contaminated regions. The subjective performance evaluation is conducted on the basis of a visual inspection of the visualized detection locations. The quantitative analysis is performed on an image fragment, containing manually derived visible noise-pattern pixel locations.

The images from the dataset are JPEG compressed with a typical setting of $Q = 25$, leading to a PSNR interval of 25–35 dB, which are supplied as input to the proposed detection systems. We apply JPEG compression in order to benchmark our results with the results obtained by [123]. The JPEG coding noise is also based 8×8 DCT compression as in MPEG. The results will therefore be comparable. Figure 6.16 and 6.17 display in red color the obtained detected potential artifact locations and in yellow the diamond-shaped expansion of this location information. Figure 6.18 and 6.19 display in the same way the detected artifact locations, for higher quality JPEG-compressed images with ($Q = 50$), leading to a PSNR interval of 28–38 dB. From Figures 6.16, 6.17, 6.18 and 6.19, the following observations are made. A detection kernel of 11×7 pixels constructed from a fixed block size of 3×3 pixels, follows the dominant edges in the picture reasonably well, covering detailed as well as coarse edges, see column (b) in the aforementioned figures. The detected activity regions (red colored pixels) show discontinuities along the object borders indicating a lower detection coherence, which are effectively appended by the diamond-shaped expansion (yellow colored pixels) of the detection signal. The expansion algorithm performs similarly as a morphological dilation operator. This is noticeable from the fact that the red-color pixel locations are surrounded by the yellow-color pixels. Furthermore, the columns (c) and (d) from the discussed figures refer to larger detection kernel sizes (19×11 and 21×13 pixels), leading to a thicker region of detected pixels (red). However, the consistency of the detected border degrades, compared to the 3×3 pixel block detection kernel.

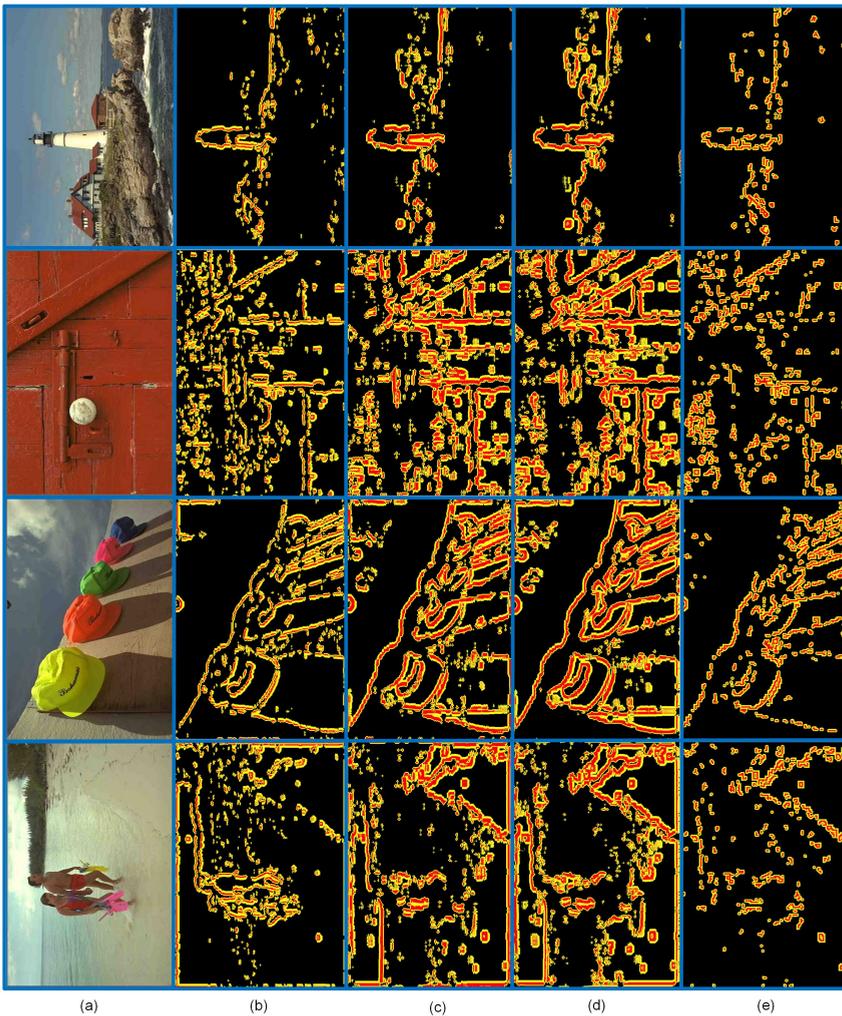


Figure 6.16 — Visualization of detected visible noise-pattern locations in JPEG-compressed pictures with $Q = 25$. In red color, detected pixel locations. In yellow color, pixel locations appended by the diamond-shaped expansion of the detection signal. (a) Original picture. (b) Detected locations for spatial-domain kernel size 11×7 pixels. (c) Detected locations for spatial-domain kernel size 19×11 pixels. (d) Detected locations for spatial-domain kernel size 21×13 pixels. (e) Detected locations for frequency-domain kernel size 20×12 pixels.

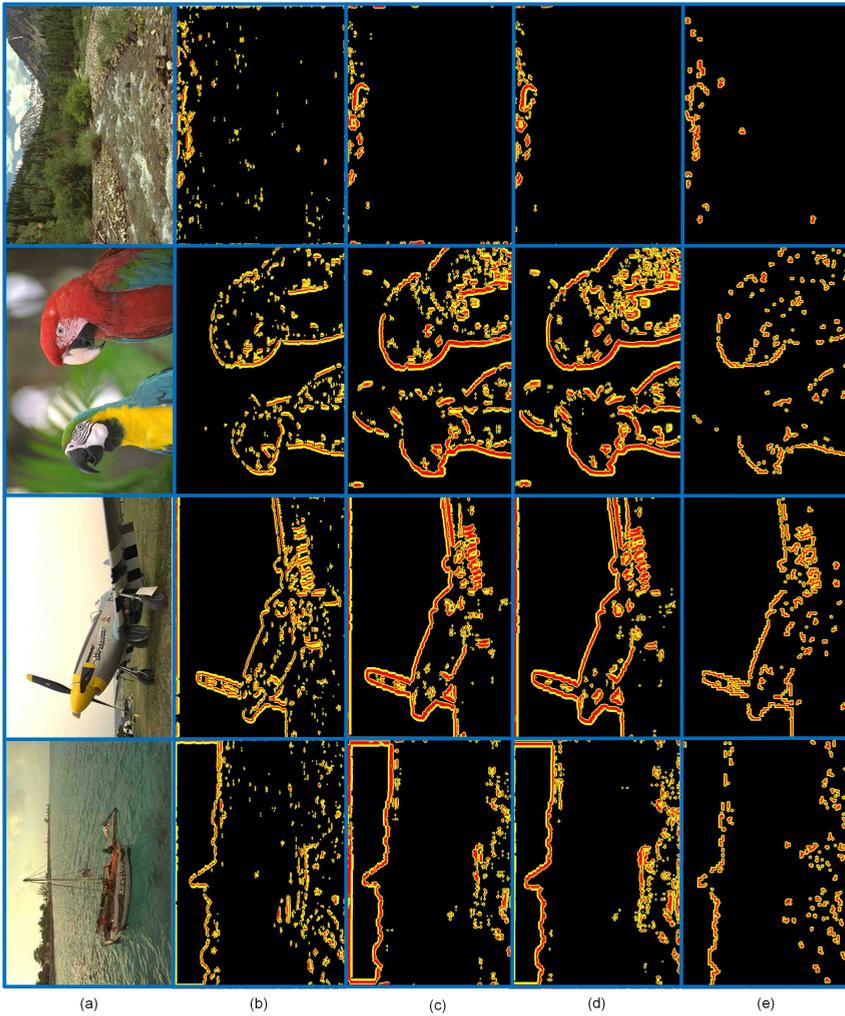


Figure 6.17 — Visualization of detected visible noise-pattern locations in JPEG-compressed pictures with $Q = 25$. In red color, detected pixel locations. In yellow color, pixel locations appended by the diamond-shaped expansion of the detection signal. (a) Original picture. (b) Detected locations for spatial-domain kernel size 11×7 pixels. (c) Detected locations for spatial-domain kernel size 19×11 pixels. (d) Detected locations for spatial-domain kernel size 21×13 pixels. (e) Detected locations for frequency-domain kernel size 20×12 pixels.

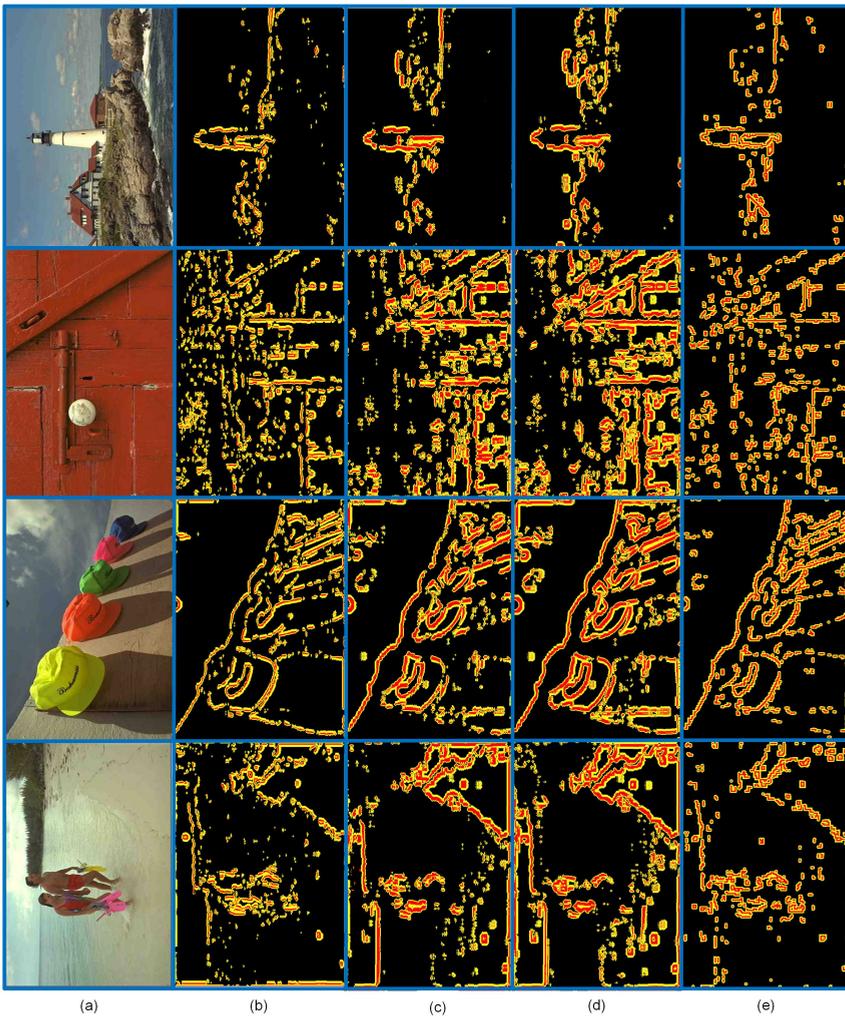


Figure 6.18 — Visualization of detected visible noise-pattern locations in JPEG-compressed pictures with $Q = 50$. In red color, detected pixel locations. In yellow color, pixel locations appended by the diamond-shaped expansion of the detection signal. (a) Original picture. (b) Detected locations for spatial-domain kernel size 11×7 pixels. (c) Detected locations for spatial-domain kernel size 19×11 pixels. (d) Detected locations for spatial-domain kernel size 21×13 pixels. (e) Detected locations for frequency-domain kernel size 20×12 pixels.

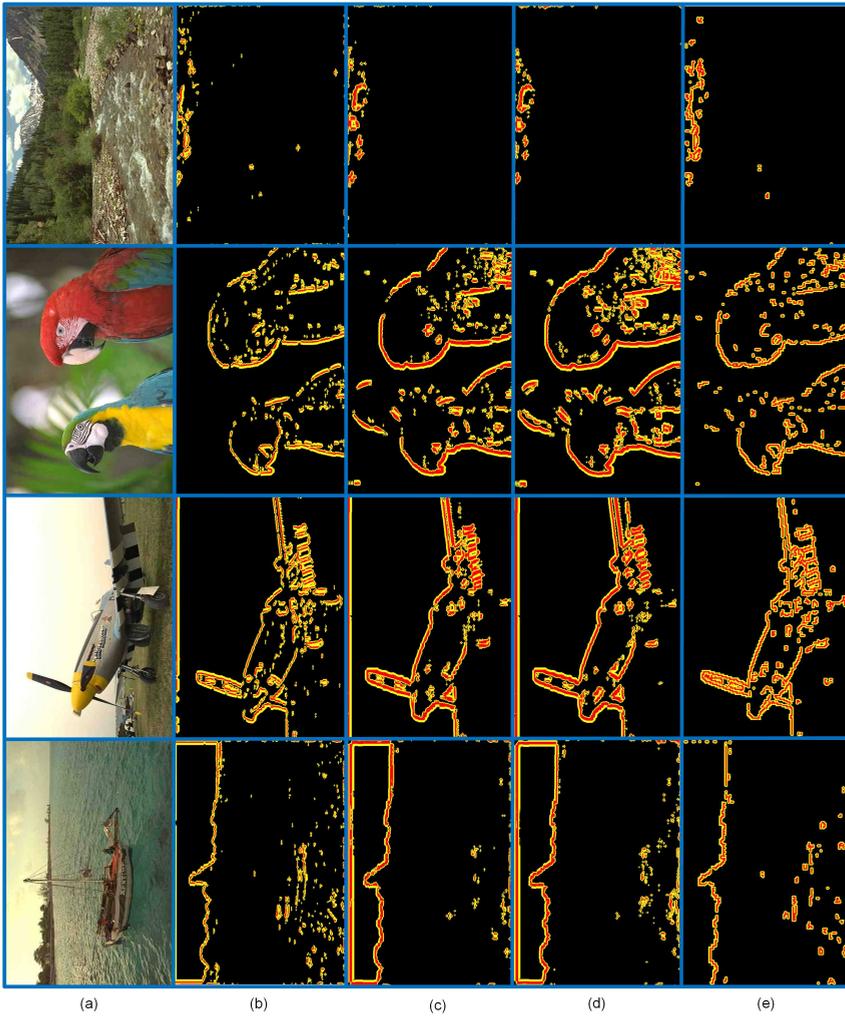


Figure 6.19 — Visualization of detected visible noise-pattern locations in JPEG-compressed pictures with $Q = 50$. In red color, detected pixel locations. In yellow color, pixel locations appended by the diamond-shaped expansion of the detection signal. (a) Original picture. (b) Detected locations for spatial-domain kernel size 11×7 pixels. (c) Detected locations for spatial-domain kernel size 19×11 pixels. (d) Detected locations for spatial-domain kernel size 21×13 pixels. (e) Detected locations for frequency-domain kernel size 20×12 pixels.

A small counter effect is that in low-detail regions, the small block detection disappear. In a subjective comparison, we have found that the last effect is less visually important than the edge consistency. In all cases, column (e) shows that the frequency-based detection of detailed edges is performing poorly with respect to consistency. The root cause of this lower performance is that the frequency-based detector considers the pre-defined noise patterns instead of the well-defined SAD metric, which clearly reveals significant changes in texture.

In a further experiment with a small quantitative analysis, we study the actual objective detection performances for the various detection kernels, using a single typical image fragment from the dataset. The selected fragment is part of a larger region, which has been identified by an expert panel [123] to contain visible coding artifacts. This identification has been adopted as ground truth (indicated as green-color pixels) in our experiment, see Fig. 6.20(b). In the following subfigures we first introduce coding noise by JPEG compression with a fixed quality setting of $Q = 25$, except subfigures (a) and (c). The purpose of the experiment is to measure the detection score of our proposed algorithm (without and with expansion) of the ground-truth area, that was indicated by the expert panel. Figures 6.20(c) and (d) are only added for exaggerated visualization of the coding noise for two settings $Q = 50$ and $Q = 25$, respectively. These subpictures show that the critical coding artifacts are along the border of the object and these artifacts appear with similar strengths for both quantizer settings, whereas the other coding noise grows with the coarseness of the settings. Our approach targets the detection of this constantly visible coding noise.

Figures 6.20(e),(f),(g) and (h) depict in red color the detected noise-pattern pixel locations (partly overwriting the ground truth). From the combined overlay, it becomes clear that the frequency-domain detection (in (h)) outperforms the spatial-domain detection (in (e) and (f)), with respect to detection consistency (red color) along the object border. The detection depicted in (g) shows a detection consistency, however, this detection is achieved in the “flat and/or low-frequency” region and partly in the noise-pattern location.

As a next step, we invoke the diamond-shaped expansion of the detected data, so that the discontinuities along the object border are filled. This expansion process is applied to both spatial-domain and frequency-domain detection, as shown in Figs. 6.20(e)(f)(g) and (h), respectively. Except for the situation depicted in Fig. 6.20(g), the expansion algorithm works properly and creates a consistently filled area along the border. In all subpictures, the ground-truth regions are overwritten by first the red detected pixels, which are expanded with the blue pixels from the expansion. However, in Fig. 6.20(g), there remains a gap between the expanded region and the actual object border, visible as remaining green pixels from the ground truth. From these subfigures, it can be observed that this two-step detection approach results in a detection coverage

Table 6.8 — *Detection score performance in percentages of noise-pattern pixel detection in the spatial domain and frequency domain for native SD resolution video. The kernel and block sizes are indicated in pixels. The table corresponds to the subfigures (e) (f) (g) and (h) in Fig. 6.20. True detected pixels are red colored in corresponding visual subfigures and expansion pixels are blue colored in those subfigures.*

	Spatial-domain kernel and block size			Frequency-domain kernel and block size
kernel	11×7	19×11	21×13	20×12
block size	3×3	mixed	5×5	4×4
JPEG Q=25				
True detected	22%	48%	9%	72%
Expansion detected	58%	40%	52%	27%
Total	80%	88%	62%	99%
JPEG Q=50				
True detected	34%	54%	14%	63%
Expansion detected	52%	35%	56%	35%
Total	86%	89%	70%	98%

of the ground truth of nearly 100% for the frequency domain. For the spatial domain, the detection coverage depends on the detection-kernel aperture and shows a substantial noise-pattern pixel coverage. However, this coverage declines for a detection kernel based on fixed 5×5 pixel blocks. There is also a counter effect. A larger detection kernel having a larger block size, produces a broader area along the object border, covering also non-contaminated pixels. This also holds for the frequency-domain approach. We will later verify the perceptual quality of this aspect. To quantify the detection score, we have measured the pixel coverage of this experiment. The pixel locations indicated in green represent the annotation from the expert panel. A 100% detection score means that all annotated green pixels have been successfully detected. The counting only considers the detection of the green pixels, including the expansion algorithm. The detection score results are depicted in Table 6.8, which indicates the percentages of detected noise-pattern pixels for all four previously discussed detection kernels (three different spatial-domain kernels and one frequency-domain kernel). The detailed numbers reflect the percentages of the red and blue pixels compared to the green ground-truth pixels. For similar detection results on full-HD upscaled video, see Table E.1 in Appendix E. Table 6.8 indicates that the detection score of frequency-domain detection is nearly 100% compared to the spatial-domain 11×7 detection kernel, which

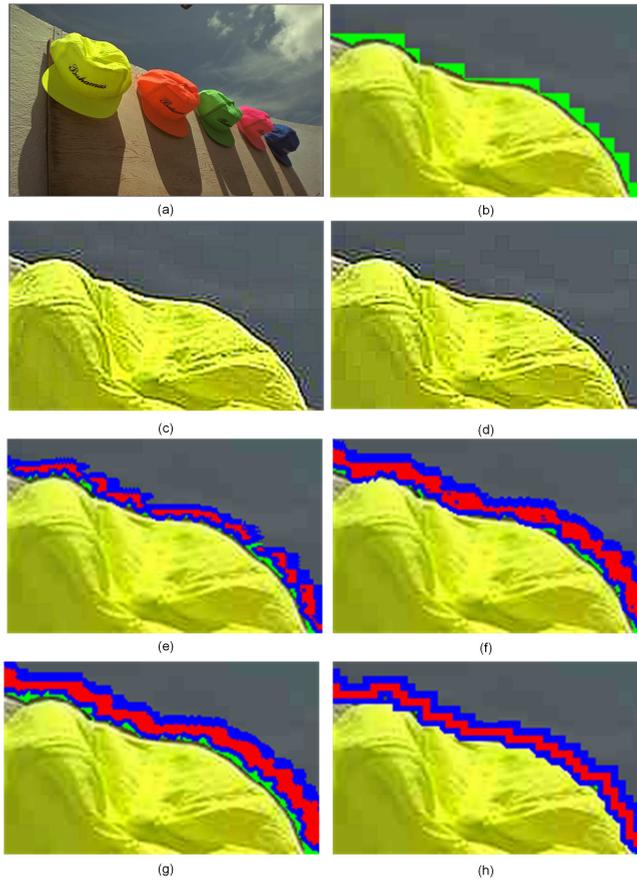


Figure 6.20 — Visible coding-noise locations from 8×8 DCT-based JPEG compression with ground truth from an expert panel. All images have setting $Q = 25$, except subfigures (a) and (c). (a) Original picture. (b) Zoomed image region with manually annotated ground-truth noise locations indicated by green pixels. (c) and (d) Sharpened image region with noise locations from JPEG compression with $Q = 50$ and $Q = 25$, respectively. Sharpening is only for visualization. (e), (f) and (g) In red color, spatial-domain detected noise pixels based on a kernel aperture of 11×7 , 19×11 and 21×13 pixels, respectively. (h) In red color, frequency-domain detected noise pixels based on a kernel aperture of 20×12 pixels. In the four bottom subfigures, the blue color indicates the diamond-shaped expansion of the detection signal.

has a pixel-based detection score of 80%. For this reason, we have employed increased detection-kernel apertures in the spatial domain by increasing the block size in either uniform or in mixed form. However, the true detection scores obtained for these larger kernel sizes in the spatial domain are still lower than obtained with the frequency-domain approach. The reason for this performance difference is twofold. First, when increasing the detection-kernel block size, the detection efficiency declines due to an SAD saturation effect. This effect occurs when computing the SAD values over larger textured areas. Already with small block sizes, the SAD value is quite high, so that for larger blocks this value further increases and therefore loses its discriminative capability. Second, when constructing a detection kernel based on a fixed large block size of 5×5 pixels, the detection becomes already vertically active prior to the actual noise-contaminated region, resulting in a detection offset. Similarly this also occurs in the horizontal direction although it starts later. As a result, the actual detection area is shifted away from the actual object border and thus also from the coding noise (this makes the green pixels not fully detected). This shifting happens despite the good detection consistency. The described phenomenon is probably due to the fixed threshold setting for spatial-domain processing.

For the spatial-domain detection kernels, the diamond-shaped expansion in combination with the detected noise-pattern pixels results in a total noise-pattern pixel detection varying between 62% and 88%, while for the frequency domain this detection becomes nearly 100%. This is visualized by diminishing the green-colored pixels. In the former examples, also non-contaminated pixels may obtain a filter strength value due to the diamond-shaped expansion. The impact of this incorrect filter-strength assignment is limited due to: (1) a decline of the filter-strength control signal and (2) protection from the adaptive low-pass filter, which facilitates an integrated edge protection and dynamic filter-aperture control.

C. Coding noise reduction performance

In the third and last experiment, we will not only detect the coding noise, but also apply the corresponding low-pass filter, as described in Section 6.4.3 and depicted in Fig. 6.13. On the basis of the derived artifact-location signal, a locally-adaptive low-pass filter is controlled, attenuating the visible noise patterns. In this experiment we measure the visual enhancement both locally along the border of the objects and globally at picture level. The local measurement is performed on the basis of the noise-detected pixels only. Both measurements are based on the Peak Signal-to-Noise Ratio (PSNR). The reference signal is the uncoded original image, which is corrupted by coding noise from the JPEG compression. Again, we apply JPEG compression in order to benchmark our results with the results obtained by [123]. Besides objective PSNR measurements, we also discuss the subjective performance.

Table 6.9 — *Local PSNR measurement after adaptive low-pass filtering (JPEG compression with quantizer setting $Q=50$).*

Picture	Spatial domain						Frequency domain	
	kernel size 11×7		kernel size 19×11		kernel size 21×13		kernel size 20×12	
	<i>orig.</i>	<i>LP.</i>	<i>orig.</i>	<i>LP.</i>	<i>orig.</i>	<i>LP.</i>	<i>orig.</i>	<i>LP.</i>
beach	35.67	35.99	36.50	36.26	37.24	37.26	34.88	35.10
caps	35.69	36.11	37.03	37.05	37.78	37.80	35.51	35.88
door	34.39	34.51	35.67	35.35	36.46	36.33	34.24	34.21
lighthouse	34.38	34.62	35.97	35.91	36.76	36.68	34.17	34.56
parrots	36.21	36.33	37.65	36.80	38.33	37.85	36.59	36.31
plane	34.90	35.33	35.77	36.12	36.74	37.09	34.31	34.83
sailboat	35.32	35.57	35.99	35.86	36.42	36.42	33.35	33.76
stream	33.47	33.70	34.41	34.72	35.66	35.81	33.81	34.17
Analysis								
average	35.00	35.27	36.12	36.01	36.92	36.91	34.61	34.85
delta	-	+0.27	-	-0.11	-	-0.01	-	+0.24

Let us start with a PSNR analysis of the four kernels, both for local and global measurements. Hereby, the local PSNR indicates the quality improvement obtained due to locally-adaptive low-pass filtering and is calculated on the basis of all detected pixels, which also form the area where the filter-strength control signal is active. The global PSNR is calculated on a per-picture basis, revealing the overall impact of the locally-adaptive low-pass filtering. Table 6.9 indicates the local PSNR derived on the basis of the area formed by the red- and yellow-colored pixels, see Figs. 6.18 and 6.19. Table 6.10 indicates the local PSNR derived on the basis of the red- and yellow-colored pixels, see Fig. 6.16 and 6.17. Note that due to the fact that each detection kernel selects a different set of reference pixels, the original local PSNR differs for equal images and depends on those reference pixels. The local PSNR results for JPEG-compressed pictures with $Q = 50$ are depicted in Table 6.9. From this table it becomes immediately clear that there are two detection systems providing on the average an increase of the PSNR. The first system operates in the spatial domain with the smallest kernel of 11×7 pixels and 3×3 pixel blocks, while the second system operates in the frequency domain with a detection aperture of 20×12 pixels and is based on 4×4 pixel blocks. The spatial-domain system provides an improvement for all pictures in the dataset, whereas the frequency-domain system provides an

improvement for the majority of pictures in the dataset. It is remarkable to observe that the frequency-domain detection system operates on a set of pixels, which on average have the lowest PSNR, when compared to the other detection systems. Artifact-location detection in the spatial domain, which operates with a detection-kernel aperture of 19×11 or 21×13 pixels, on average deteriorate the image quality. This variation in performance is caused by the fact that JPEG-compressed images with $Q = 50$ have a high picture quality and the total detected artifact locations should be carefully filtered with constrained filter settings. Although the frequency-domain detection gives the highest detection score (see previous subsection), the PSNR results are somewhat lower than the small kernel spatial-domain processing. This difference is explained by the broader detection region around object borders, which is sometimes expanding outside the ground-truth area. The corresponding filtering therefore lowers the measured image quality in those expanded regions. Table 6.10 reveals the local quality improvement, for the JPEG-compressed pictures with a coarser quantizer value of $Q = 25$ depicted in Figs. 6.18 and 6.19. From this table, it can be observed that on average all detection systems provide a local enhanced PSNR. For strongly compressed image data, composed of large flat regions such as in the caps and plane images, the detection system operating in the frequency domain outperforms the spatial domain with a local PSNR improvement of up to 0.5 dB. It is remarkable that the spatial-domain detection with a detection kernel of 11×7 pixels operates on a set of pixels, which on average have the lowest PSNR, when compared to the other detection systems. This makes the chances for visual enhancement higher.

In Tables 6.11 and 6.12, the global PSNR is presented. These results are fully consistent with the local measured PSNR values. For low-quality compression, all kernels provide a small improvement of the PSNR, while for high-quality compression only the spatial-domain small kernel and the frequency-domain system provide an on the average positive PSNR enhancement.

On the basis of the previous results, it is concluded that artifact-location detection conducted in the spatial domain employing a detection kernel of 11×7 pixels, based on a fixed block size of 3×3 pixels provides good results on picture quality enhancement for a broad range of image content and compression settings. Embarking on the previous results with PSNR, we now subjectively compare the picture quality between the originally decompressed and the locally-adaptive low-pass filtered version. We conduct this visual comparison only for the best performing detection kernel, i.e. the spatial-domain detection kernel with 11×7 pixels using a block size of 3×3 pixels.

Table 6.10 — Local measured PSNR after adaptive low-pass filtering (JPEG, Q=25).

Picture	Spatial domain						Frequency domain	
	kernel size 11 × 7		kernel size 19 × 11		kernel size 21 × 13		kernel size 20 × 12	
	<i>orig.</i>	<i>LP.</i>	<i>orig.</i>	<i>LP.</i>	<i>orig.</i>	<i>LP.</i>	<i>orig.</i>	<i>LP.</i>
beach	31.74	32.04	33.09	33.44	34.27	34.50	32.31	32.59
caps	32.10	32.45	33.64	34.00	34.99	35.31	33.05	33.57
door	31.42	31.58	32.80	32.83	33.84	33.92	31.91	31.96
lighthouse	30.82	31.03	32.64	32.86	34.25	34.42	32.00	32.42
parrots	33.01	33.18	34.77	34.66	35.75	35.77	34.15	34.15
plane	31.33	31.57	32.28	32.62	34.02	34.35	31.82	32.37
sailboat	30.85	31.11	32.25	32.58	33.50	33.65	31.17	31.50
stream	29.44	29.54	30.68	30.92	32.58	32.72	31.37	31.62
Analysis								
average	31.34	31.56	32.77	32.99	34.15	34.33	32.22	32.52
delta	-	+0.22	-	+0.22	-	+0.18	-	+0.30

Table 6.11 — Global measured PSNR after adaptive low-pass filtering (JPEG, Q=25).

Picture	<i>orig.</i>	Spatial domain			Frequency domain
		kernel size			
		11 × 7	19 × 11	21 × 13	20 × 12
		<i>LP.</i>	<i>LP.</i>	<i>LP.</i>	<i>LP.</i>
beach	33.27	33.37	33.46	33.35	33.31
caps	33.87	33.98	34.02	33.96	33.98
door	32.84	32.89	32.85	32.88	32.85
lighthouse	30.07	30.09	30.09	30.08	30.09
parrots	35.39	35.42	35.34	35.39	35.39
plane	32.56	32.62	32.66	32.61	32.64
sailboat	29.46	29.49	29.51	29.48	29.47
stream	25.75	25.76	25.76	25.76	25.76
Analysis					
average	31.65	31.70	31.71	31.69	31.69
delta	-	+0.05	+0.06	+0.04	+0.04

Let us start with a visual inspection of image fragments depicted in

Table 6.12 — *Global measured PSNR after adaptive low-pass filtering (JPEG, Q=50).*

Picture		Spatial domain			Frequency domain
		kernel size			
		11×7	19×11	21×13	20×12
	<i>orig.</i>	<i>LP.</i>	<i>LP.</i>	<i>LP.</i>	<i>LP.</i>
beach	35.73	35.79	35.64	35.73	35.76
caps	36.15	36.24	36.16	36.16	36.22
door	34.77	34.80	34.62	34.73	34.77
lighthouse	32.24	32.26	32.24	32.24	32.26
parrots	37.72	37.74	37.43	37.59	37.68
plane	34.76	34.82	34.82	34.80	34.84
sailboat	31.85	31.87	31.84	31.85	31.87
stream	28.10	28.10	28.10	28.10	28.10
Analysis					
average	33.92	33.95	33.86	33.90	33.94
delta	-	+0.03	-0.06	-0.02	+0.02

columns (a) and (b) of Fig. 6.21. For visual clarity, the image fragments have been zoomed 300%, such that the coding noise can be clearly observed. All image fragments depicted in columns (a) and (c) of Fig. 6.21 show triangular-shaped noise patterns introduced by to JPEG compression. Columns (b) and (d) show for all fragments a clear reduction of the visible coding noise. However, due to the limited vertical aperture of the detection kernel, not all noise-pattern pixels are assigned a sufficient filter strength. As a result, the adaptive low-pass filter is not able to sufficiently smooth these pixels, resulting in some remaining visible distortion after low-pass filtering. This type of visible distortion is particularly annoying due to the fact that it occurs at a distance of up to 7 pixels from the edge/texture transition. The third image fragment from the top of column (b) shows a part of the parrots pecker. At the right side of that image, slightly above the middle, there is a small discontinuity in the green background due to insufficient low-pass filtering. Note that this distortion is not present in the filtered version depicted in column (d), due lower quantization and thus less severe distortion, requiring less intensive low-pass filtering.

D. Comparison with existing solution

In this subsection the proposed detection systems are visually compared with the results from [123], which was published in the same time frame. For this comparison, we have selected four typical images from the applied dataset.

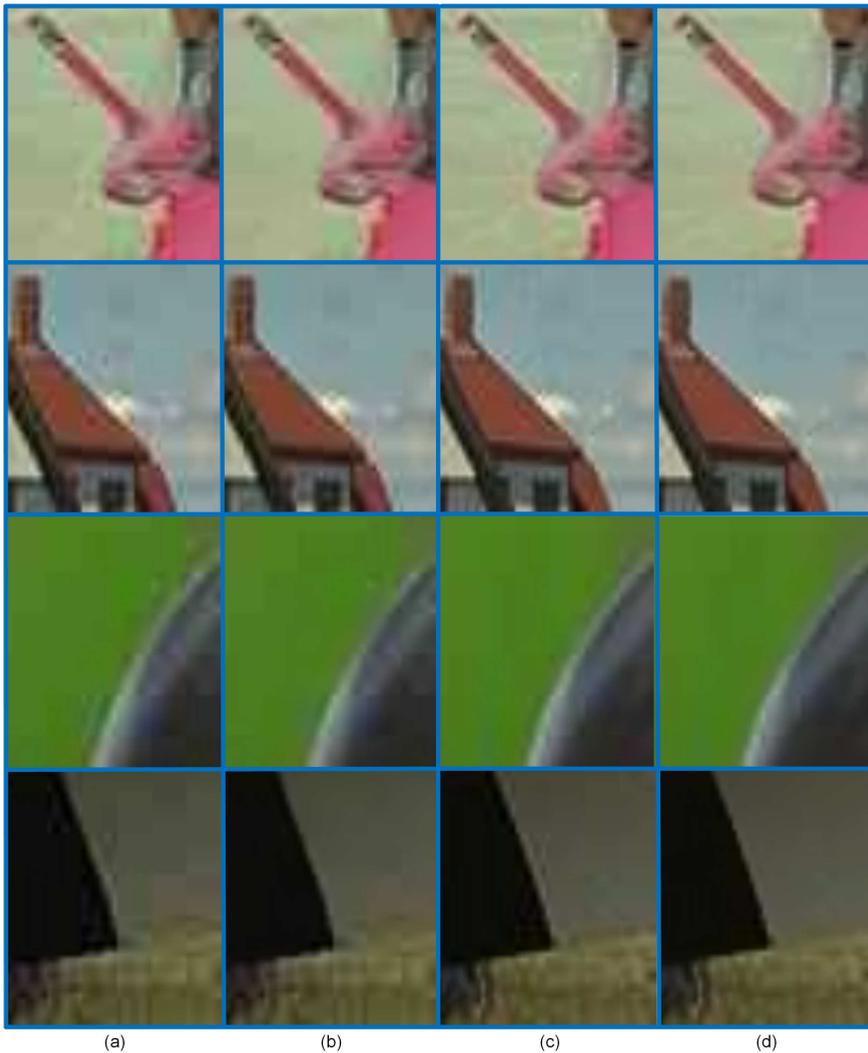


Figure 6.21 — Image fragments showing the effect of noise detection and subsequent locally-adaptive low-pass filtering for JPEG-compressed images with $Q = 25$ for columns (a) and (b) and $Q = 50$ for columns (c) and (d). Column (a): zoomed 300% JPEG-compressed image fragment. (b): zoomed detection and filtered JPEG-compressed image fragment. (c): zoomed 300% JPEG-compressed image fragment. (d): zoomed detection and filtered JPEG-compressed image fragment.

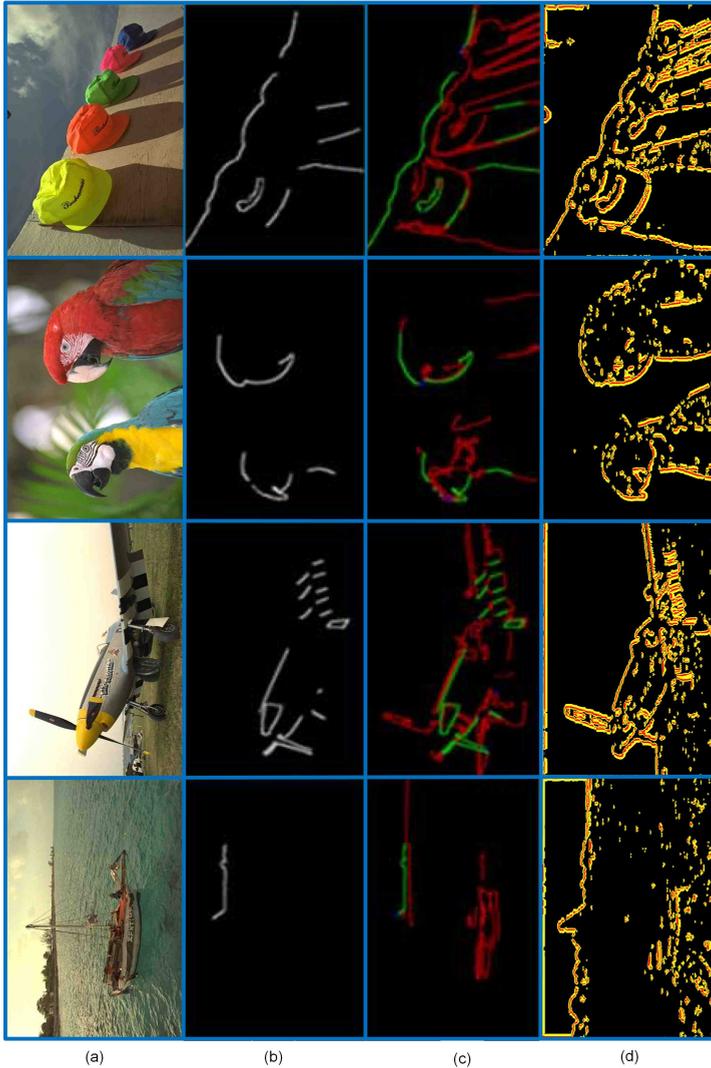


Figure 6.22 — Visual comparison of subjective ringing regions of [123] and our small-kernel spatial-domain detection results for JPEG-compressed images with $Q = 25$. Column (a): Original pictures. (b): Subjective ringing regions indicated by expert panel taken from [123]. (c): Detection results obtained with Canny edge detection, non-linear smoothing and bilateral filtering (taken from [123]). (d): Detected locations for spatial-domain kernel size 11×7 pixels.

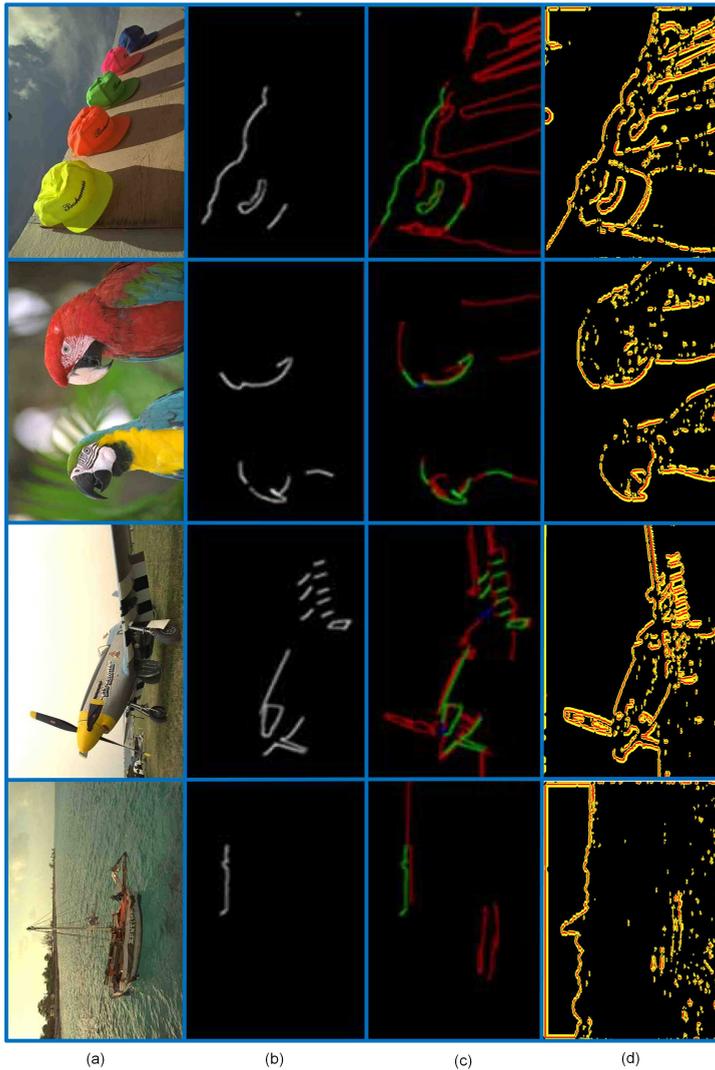


Figure 6.23 — Visual comparison of subjective ringing regions of [123] and our small-kernel spatial-domain detection results for JPEG-compressed images with $Q = 50$. Column (a): Original pictures. (b): Subjective ringing regions indicated by expert panel taken from [123]. (c): Detection results obtained with Canny edge detection, non-linear smoothing and bilateral filtering (taken from [123]). (d): Detected locations for spatial-domain kernel size 11×7 pixels.

Figure 6.22 depicts this comparison in detail for the selected images. Column (b) shows the identified object borders manually indicated by the expert panel. Column (c) shows the automatic detection performance obtained by special processing. The author indicates in [123] that this automatic processing involves the following steps: Canny edge detection (without its inherent smoothing step), applied to a non-linearly smoothed image using a bilateral filter to obtain the perceptually more meaningful edges. The detected edge pixels are necessarily combined into perceptually salient elements, facilitating further analysis and processing. The red color indicates the additionally found object borders from these processing steps. In column (d), the detection results of our proposed 11×7 spatial-domain kernel are shown. It should be noticed that our small detection blobs do not result in severe filtering, because they are only one or two pixels in size and the filtering has a diamond-shaped decline of the filter control.

There are two major differences between the literature and our proposal. First, with respect to detection, our solution covers all essential object borders over the full object shape. This is not the case in the published reference work. In that work, parts of the object border are missing, with the risk that filtering is not applied despite the noise visibility. Second, the proposed detection processing from literature is significantly more complex than our spatial-domain processing. Therefore, we consider our proposal more attractive for implementation in a real television system.

6.6 Conclusions

In this chapter, we have proposed two block-based artifact-location detection systems for detecting the locations of mosquito noise and ringing noise patterns. Our proposed solutions operate either in the spatial domain or frequency domain and apply an artifact-detection kernel, which is constructed using either fixed-sized or variable-sized blocks. For each block constructing the detection kernel in the spatial-domain processing, the SAD value is calculated as an activity metric. For the frequency-domain solution, a set of noise and ringing patterns are defined for detection. The activity of the detected noise or ringing is classified in three simple indications, like “flat and/or low-frequency”, “noise contaminated (mosquito noise and ringing)” and “texture”.

The usage of these signal features is employed in two ways. First, on the basis of the block-based signal classification, a simplified video model is derived, using the surrounding blocks of the considered area and classify those blocks in a similar way. These surrounding blocks act as additional context information suitable for context reasoning about the visibility of possible coding noise. This gives a more robust location detection of potential noise-contaminated regions. Second, these block-based signal features are also employed after-

wards to expand the detection signal employing a diamond-shaped aperture. In this way, the detection signal becomes more consistent as a filter control signal and increases the area coverage of the detected contaminated region. The derived signal controls a locally-adaptive low-pass filter, attenuating the located noise patterns. Using this new artifact-location detection concept, we have found the following performance aspects.

Performance of artifact detection. The first performance measurement on potential artifact-location detection is obtained with the spatial-domain detection system, employing a detection kernel of 11×7 pixels based on a fixed block size of 3×3 pixels. The pixel detection associated with the noise patterns, in combination with diamond-shaped expansion, has a score of 80 and 86%, for JPEG-compressed pictures with $Q = 25$, $Q = 50$, respectively. We have also discussed a frequency-domain detection system, employing a detection kernel of 20×12 pixels based on a fixed block size of 4×4 pixels. This detection provides a noise-pattern pixel detection in combination with a diamond-shaped expansion. The detection score results in 98 and 99%, for JPEG-compressed pictures with $Q = 50$, $Q = 25$, respectively.

Although the detection score obtained in the frequency domain is higher, the performance of the spatial-domain detection in conjunction with expansion, is more accurate along a broad range of object contours including fine detailed edges. Furthermore, the 2D expansion is more subtle due to the smaller block size and the pixel-based detection, resulting in a limited amount of false-positive pixel detections.

Local image enhancement. Visible coding-noise reduction obtained on the basis of a spatial-domain detection system, operating with a kernel aperture of 11×7 pixels, results in an on the average local image enhancement of 0.2 and 0.3 dB for JPEG-compressed pictures with $Q = 25$, $Q = 50$, respectively. Local image enhancement on the basis of a frequency-domain kernel of 20×12 pixels, results in an on the average improvement of 0.3 and 0.2 dB, for JPEG-compressed pictures with $Q = 25$, $Q = 50$, respectively. We have found that for modest coding quality images, the enhancement on average obtained with frequency-domain detection is better compared to the spatial domain, albeit with sometimes significant variations per individual video sequence. In contrast and for the same image quality, the spatial-domain detection always results in a positive enhancement. For high coding quality, the spatial-domain detection outperforms the frequency-domain system on average and always contributes to a positive local enhancement, which is due to a more subtle artifact-location detection.

Final performance differences. Software-based simulation of the combination employing both detection and filtering has revealed the following differences. The

spatial-domain solution employing a kernel of 11×7 pixels provides an on average global increase in PSNR of +0.05 and +0.03 dB for all images from the applied dataset, which are JPEG-compressed with $Q = 25$, $Q = 50$, respectively. For the images of this dataset, the frequency-domain detection system using a kernel of 20×12 pixels, provides a global image enhancement of +0.04 and +0.02 dB for JPEG-compressed images with $Q = 25$, $Q = 50$, respectively. From these experiments, we conclude that both systems slightly enhance the image quality at a global scale, but with locally strongly improved areas. Furthermore, the combined system avoids excessive image blur from filtering, e.g noticeable from the positive PSNR contributions, which is due to the context reasoning in the kernel area and the detailed decision making.

Hardware realization in real commercial TV systems. Due to the successful performance of the spatial-domain detection system and the low implementation costs, this detection system has been adopted for embedding in a real digital TV platform. The embedding was realized in two chips, called SX6 and SX7, targeting DTV reception. Furthermore, this detection system has been adopted as part of a back-end video processor chip FRCX¹, aiming at Ultra HD frame-rate conversion.

¹All these chips are commercially available from Sigma Design, USA.

Conclusions

7.1 Conclusion of the individual chapters

Chapter 2. This chapter provides a brief overview on parts of the MPEG standards relevant for this thesis. Moreover, a brief introduction is given on noise-patterns introduced by MPEG video compression and on intra-program video navigation for personal video recording. Furthermore, conventional video navigation is discussed, especially the limitations of this navigation form.

Chapter 3. This chapter aims at solutions for networked navigation through MPEG-2-compressed databases. To this end, we have classified video navigation into three categories, of which two categories are characterized by a networked decoder, operating at a distance from a storage device elsewhere in the network. The third category involves a different navigation concept and is presented in Chapter 4. The two proposed video navigation categories from this chapter are suitable to be employed in a client-server-based communication system. Both solutions feature communication interoperability, based on standard MPEG coding techniques and coded MPEG information transmitted across the network. The two proposed solutions are full-frame based navigation techniques, the first for fast-search and slow-motion playback and the second for hierarchical screen searching.

The networked fast-search and slow-motion video navigation is based on (1) re-used intra-coded MPEG-compressed normal-play video pictures for deriving the fast-search navigation sequence, and (2) full re-use of all normal-play pictures for the slow-motion sequence. In order to adapt the playback speed, frame rate, bit rate and field rendering control, we employ artificially generated repetition pictures, which repeat normal-play reference pictures. This adaptation is achieved by assuming a fixed bit rate and then calculating the transmission time for each re-used intra-coded picture.

The proposed networked hierarchical mosaic-screen video navigation is based on hierarchical mosaic screens involving the usage of MPEG-2-compressed subpictures derived from intra-coded normal-play pictures during record-

ing. Flexibility in the composition of mosaic screens is essential for providing different temporal instant overviews. This is obtained by coding each subpicture in a fixed bit cost, involving fixed-cost coded “*mini slices*” on the basis of P-type coding syntax, thereby facilitating easy retrieval from the storage device, while simplifying the construction of the final mosaic screen with the compressed subpictures.

Chapter 4. Video navigation is further extended with a new audiovisual proposal, involving multiple information signals (2 video and 1 audio) to improve the perception of navigation playback. The method encompasses re-use of MPEG-2-compressed normal-play video fragments with corresponding audio information in combination with an additional fast-search video navigation window. The normal-play video fragments are presented in a primary window enhanced with an auditive signal, whereas the fast-search video navigation is presented in a smaller secondary Picture-in-Picture (PiP) window, overall resulting in a dual-window video navigation solution. The algorithms for constructing the navigation signal involve scalable MPEG decoding and re-encoding of intra-coded normal-play pictures, forming the fast-search information during recording which is stored as metadata. Moreover, the re-used normal-play fragments are processed for amongst others audio padding adaptation and removal of normal-play predicted coded-pictures without available reference. The proposed algorithmic simplifications for an MPEG-2 and H.264/MPEG4-AVC intraframe decoding result in pictures of good subjective video quality, while the objective quality in terms of PSNR is low 28.69 dB for MPEG-2 and 26.30 dB for H.264/MPEG4-AVC. For video navigation, this quality is sufficient as the viewer will have insufficient time to visually inspect the individual images.

As a general conclusion, the three video navigation solutions from this thesis address each a particular navigation time interval. The proposed concepts show a high commonality and differ mainly in the presentation of video navigation information. A key aspect of this work is the re-use of normal-play encoded audiovisual information, involving specific processing in the MPEG-2 compressed domain, resulting in a compliant video navigation signal. This enables re-use of video and audio decoding components, in such a way that transcoding is avoided and the additional processing can be embedded on the existing processing units and control CPU. Furthermore, derived CPI information is re-used by all three video navigation methods. This even holds for the navigation solutions which employ subpictures. By separating the navigation processing in a recording stage and a playback stage, computational complexity can be controlled. We conclude that on the basis of the chosen concept with small-picture generation during recording and the associated metadata, the re-use of already coded pictures with scalable and/or partial decoding and

the measures for complexity control, all together enable the combination of the proposed PVR concepts to create a framework that can handle short-, medium- and long-time interval video navigation. Moreover, this framework would be feasible and realizable with limited complexity.

Chapter 5 MPEG-coded information is transmitted over terrestrial channels for mobile reception, making use of IP encapsulation in the framework of DVB-H standard. The combination of both standards does not achieve the best possible robustness. This chapter presents an improved DVB-H link layer, capable of improving the robustness and providing a best-effort signal degradation, while minimizing data communication. Key aspects of the solution are the locally obtained reliability and location information, revealing the reliability status of individually received bytes before and after error correction and the storage locations of correctly received IP datagrams. The MPEG-received reliability information is elegantly employed in two ways. First, this reliability information is applied for error and erasure decoding by the link-layer FEC stage. For the situation that after this second FEC stage an MPE-FEC frame is still incorrect, this reliability information is used for a second time, in combination with reliability information derived from the secondary FEC decoding stage, supplemented with the location information. In this way, correctly received and corrected IP datagrams are extracted from the same defect MPE-FEC frame. This leads typically to a completely corrected MPE-FEC frame without errors, while sometimes exceptions occur and errors remain. We have shown that this concept for retrieving completely correct MPE-FEC frames indeed clearly improves the robustness with approximately 50% and that the performance curves tend to cluster around the same critical performance degradation point. Moreover, IP recovery in the remaining defect MPE-FEC frames is enabled on the basis of joint reliability and correct IP datagram location information, resulting in up to 20% additional IP datagram recovery. Evidently, this performance strongly depends on the IP datagram size, as well as the error probability. Furthermore, when forwarding IP datagrams only once to the network layer, the data communication is minimized and contributes to a reduced power consumption. This improved DVB-H link layer implementation has been adopted in a commercial DVB-H receiver.

Chapter 6 MPEG-compressed video is always communicated through bandwidth-limited channels, leading to sometimes visible and even annoying coding noise. In modern digital television systems, these coding artifacts are typically detected and correspondingly attenuated, which gives undesirable image blur. We have proposed two block-based artifact-location detection systems for detecting the locations of mosquito noise and ringing noise patterns. A key aspect of both solutions is the construction of an artifact-detection kernel, which is constructed either in the spatial domain or in the frequency domain,

using either fixed-sized or variable-sized blocks. For each block constructing the detection kernel in the spatial-domain processing, the SAD value is calculated as an activity metric. For the frequency-domain solution, a set of noise and ringing patterns are defined for detection. The activity of the detected noise or ringing is classified in three simple indications, like “flat and/or low-frequency”, “noise contaminated (mosquito noise and ringing)” and “texture”. On the basis of the classified block-based signal features, a simplified video signal model is derived of the local region, suitable for context reasoning with the surrounding blocks about the visibility of possible coding noise. The block-based signal features are also employed afterwards to expand the detection signal, thereby improving the consistency of the detection signal for filter control and increasing the accuracy of the covered area of the detected contaminated region. The derived signal controls a locally-adaptive low-pass filter, which is only active in the detected region and seamlessly adapts the filtering if the certainty about noise is lower. The artifact detection score for the frequency-domain based detection is higher compared to the spatial-domain detection. However, the spatial-domain solution always provides an improvement of a local image PSNR enhancement of 0.2 and 0.3 dB for high and low compression factors, respectively. The proposed spatial-domain noise detection system, has been adopted in practice and is implemented in various commercial DTV applications.

7.2 Discussion on research questions

The proposed solutions are now addressed with respect to the posed research questions in Section 1.3.

RQ1 How to efficiently perform trick-play playback on MPEG-compressed audiovisual information in various communication situations?

For the first research question (RQ1), two chapters are relevant. In Chapter 3, we propose three algorithms for deriving a video-only navigation signal. Two algorithms address networked full-frame video navigation, while the third algorithm involves mosaic screen based video navigation. In Chapter 4, we propose an algorithm for multi-signal navigation including audio information.

RQ1a How can normal-play MPEG-compressed audiovisual information be re-used for conventional trick-play playback? Conventional trick-play playback is divided in fast-search and slow-motion playback, so that the research question is answered in two ways. First, fast-search playback on MPEG-2-compressed video data is obtained by re-using intra-coded normal-play pictures, which can be efficiently retrieved from the storage medium using the storage locations, derived during recording. In this way, fast-search video navigation can be ob-

tained for speed-up factors being a multiple of the normal-play GOP length. Second, slow-motion playback is obtained by repetitive display of all normal-play images. Hereby, a distinction is made between the video format, i.e. the interlaced or progressive format of the MPEG-2-coded program. For progressive video, slow-motion playback on MPEG-2-compressed normal-play video data is established, by inserting MPEG-2 B-type repetition pictures repeating the reference pictures, while normal-play B-type pictures are repetitively decoded. For slow-motion playback of interlaced video, the algorithm is slightly different. First, the reference pictures are repeated using B-type repetition pictures, which repeat the last rendered field, thereby avoiding motion judder. Furthermore, the normal-play B-type pictures are decoded once, to avoid motion judder. The speed-error that occurs is compensated by additional repetitions of the reference pictures.

RQ1b How to perform trick play in a client-server-based networked system setup? When conducting trick play in a networked client-server-based system, MPEG-2 offers dedicated trick-play signaling within the Packetized Elementary Stream (PES) layer to control the trick-play playback. However, this form of signaling is non-mandatory, which may result in undefined system behavior when employed. It is therefore recommended that regular MPEG-2 encoding and decoding techniques are applied, which makes a trick-play signal equal to a normal-play video sequence from MPEG-2 coding perspective. For the associated algorithms to implement such way of navigation, the re-used pictures are selected and grouped and re-formatted into a new compliant MPEG stream that can be handled by any MPEG decoder. The answer on RQ1a presents algorithms for navigation based on this principle.

RQ1c How to fulfill the bit-rate and frame-rate constraints when re-using normal-play MPEG-compressed video information? This two-dimensional problem can be reduced to a one-dimensional problem by pre-setting the bit rate to a fixed practical value, which provides a fixed upper bound to the bit-cost budget for each picture constructing the video navigation sequence. For the situation that re-used MPEG-2-compressed intra-coded pictures have bit costs that satisfy the upper-bound condition, no special action has to be taken. However, for the situation that the bit costs of the re-used MPEG-2-compressed pictures exceed this upper bound, additional transmission time is required. This is achieved by the insertion of one or more artificially predictive-coded repetition pictures, repeating the last decoded picture. As these artificially predictive-coded images have a bit cost which is far less compared to the picture bit-cost budget, the involved transmission time is extremely short. The remaining transmission time is then used for the transmission of the re-used normal-play picture.

RQ1d What are the relations and limitations of high-speed search in relation to

the MPEG-based playback navigation information? Fast-search video navigation based on re-used MPEG-2-compressed pictures is conducted using solely intra-coded pictures. In the normal-play video sequence, these pictures occur at a distance equal to the GOP length. As a result, the minimal fast-search playback speed equals the normal-play GOP length. High-speed navigation-playback speeds are therefore an integer multiple of this normal-play GOP length. When conducting fast-search playback, non-consecutive normal-play pictures are constructing a video navigation sequence. When considering a typical scene duration of 3 seconds and a GOP length of 12 pictures, the maximal video navigation playback speed becomes 75, resulting in a concatenation of a single picture from each scene. However, interpretation of the visual navigation information by the viewer involves multiple display periods. We have found a good fast-search navigation performance for video navigation playback at a speed of 25, resulting in the concatenation of 3 correlated (same scene) pictures.

RQ1e What is the impact of the employed video format in relation to trick-play playback? Fast-search video navigation on the basis of re-used MPEG-2-compressed video pictures involves picture repetition on the basis of artificially predictive-coded repetition pictures. For the situation that the video format is progressive, picture repetition results in an exact duplicate of the reference picture. When the video format is interlaced, picture repetition results in an exact duplicate of the reference picture including the inter-field motion. As a result, objects that are subject to this inter-field motion cause motion judder, which is annoying for the viewer. Motion judder is avoided when applying predictive-coded field-based repetition pictures, which duplicate the last rendered field of the reference picture, depending on the search direction, thereby avoiding the appearance of motion judder. This makes the navigation playback perceptually more pleasing.

RQ1f How can audio information contribute to the video navigation efficiency? Audio information can contribute to the video navigation efficiency, provided that the audio time-interval information is of sufficient duration. We have found that auditive information that is related with a conventional fast-search video navigation provides extra information. A video navigation sequence that is based on temporally subsampling the normal-play sequence, while selecting normal-play audiovisual fragments with a 3 second duration, provides suitable and sufficiently detailed information, which is on the average perceived as a good interval for presenting audio.

RQ1g Is there a system architecture that allows conventional as well as more advanced video navigation methods? The video navigation solutions proposed in Chapter 3 and Chapter 4 employ a PVR functional block diagram, which shows a high commonality. When separating the involved navigation signal

processing over the recording mode and navigation playback mode, conventional as well as advanced video navigation methods can be facilitated. Key aspect is the derivation of metadata describing the properties of the recorded program and additional navigation information during recording. Based on this metadata, more advanced video navigation methods can be realized at navigation playback with limited complexity. Other important aspects are re-use of MPEG-coded frames, scalable decoding of these frames matching with the desired image size, and associated smart rendering. Since all three navigation forms are based on one or more of these aspects, it is possible to design an overall PVR architecture that embeds these forms of navigation.

RQ2 How to improve the robustness of a standard DVB-H link layer while avoiding excessive load on system resources?

For the second research question (RQ2), Chapter 5 is relevant.

RQ2a How can the error recovery of the embedded RS decoder be optimized leading to improved robustness? Key aspects of the solution are the locally obtained reliability (2-bit erasure) and location information, revealing the reliability status of individually received Bytes before and after error correction and the storage locations of correctly received IP datagrams. The usage of the MPEG-received reliability information is employed in two ways. First, this reliability information is applied for error and erasure decoding by the link-layer FEC stage. For the situation that after this second FEC stage an MPE-FEC frame is still incorrect, this reliability information is used for a second time, in combination with reliability information derived from the secondary FEC decoding stage, supplemented with the location information. We have shown that this concept for retrieving completely correct MPE-FEC frames indeed clearly improves the robustness with approximately 50%, while IP recovery in the remaining defect MPE-FEC frames, enabled on the basis of joint reliability and correct IP datagram location information, further leads to 20% additional IP datagram recovery.

RQ2b How to communicate correctly received and FEC-corrected IP datagrams in a smooth communication way? DVB-H is an IP-datagram-based communication standard and uses from each IP datagram header, the length field to retrieve this IP datagram from memory. This retrieval mechanism works only properly when all received IP datagrams are correct, either due to reception or after link-layer FEC. For the situation that the link-layer FEC cannot correct all erroneously received IP datagrams, loss of IP datagrams may occur. Our proposed solution is based on deriving the storage locations of correctly received IP datagrams, enabling their retrieval from memory. However, the memory may also contain defect IP datagrams after FEC, which prevents retrieval of successive IP datagrams. When deriving reliability information during link-

layer FEC and combining this information with the reliability information and also using location information derived during data reception, this combination facilitates in the successful retrieval of FEC-corrected IP datagrams from memory. It should be noted that the remaining defect IP datagrams remain in the memory and are not passed on to the network layer. In this way, all IP datagrams are forwarded once to the network layer, which contributes to the efficiency and thus reduces power consumption.

RQ3 How to efficiently detect visible coding noise locations in MPEG-coded video with sufficient performance?

For the third research question (RQ3), Chapter 6 is relevant.

RQ3a With what methods can visible MPEG noise patterns reliably be found in the image and what are the corresponding metrics? In Chapter 6, we propose two block-based artifact-location detection systems operating either in the spatial domain or frequency domain. The proposed algorithms employ an artifact-detection kernel constructed of a set of neighboring blocks. For each block constructing the detection kernel in the spatial-domain processing, the SAD value is calculated as an activity metric. For the frequency-domain solution, a set of noise and ringing patterns are defined for detection. The activity of the detected noise or ringing is classified in three simple indications, like “flat and/or low-frequency”, “noise contaminated” and “texture”. The usage of these signal features is employed in two ways. First, on the basis of the block-based signal classification a simplified video model is derived, using the surrounding blocks of the considered area and classifying those blocks in a similar way. These surrounding blocks act as additional context information, which is suitable for context reasoning about the visibility of possible coding noise. This gives a more robust location detection of potentially noise-contaminated regions. Second, these block-based signal features are also employed afterwards to expand the detection signal, employing a diamond-shaped aperture. The combined detection performance for the spatial-domain detection system varies within 62–89 %, depending on the employed block size and amount of video compression, while for the frequency-domain detection system the detection score is 98–99 %, which only depends on the amount of video compression because the involved block size is fixed.

RQ3b How can the reliability of the detection methods be improved? Visible noise patterns are located in flat and/or low-frequency regions. For the situation that a noise-contaminated region is detected, the surrounding region is tested for a larger flat and/or low-frequency region, which confirms the visibility aspect. If this large region is indeed present, the detection is confirmed, otherwise the detection is rejected.

RQ3c How can this method be embedded in a DTV platform? The deployment of these detection systems in an embedded platform can be conducted in either a separable or non-separable approach. Hereby the activity calculation and associated context reasoning are either calculated in the same processing stage, or separated over two different processing stages. When employing this detection system in a non-separable approach, the detection is embedded in a video enhancement pipe, which requires local storage of the video signal involving line memories. Hereby, the amount of line memories depends on the vertical aperture of the detection kernel, which depends on the employed block size. The adopted systems employ a block size of either 3×3 or 4×4 pixels, which is small enough to be embedded or combined with a DCT transform of 4×4 or 8×8 pixel blocks. Second, the SAD metric is a standard operation in modern CPUs and DSPs, so that it can be easily implemented, even with parallelism. Third, the frequency transform is based on a DCT computation, which enables re-use of processing blocks in MPEG-based systems. For smooth implementation of the kernel memory, the activity calculation and the spatial reasoning are separated. The advantage of such an approach is that the vertical aperture and thus the amount of embedded line memories, depends on the employed block size and not on the deployed detection kernel size. However, this approach increases the external memory input/output for temporary storage of the video data. Such a separable system allows the detection kernel to have an increased vertical aperture, while avoiding the need for the corresponding embedded line memories.

7.3 Discussion and outlook

Video navigation

Video navigation is an essential feature, which will also be present in future digital storage systems. Due to advances in technology, these digital storage systems may differ from current solutions and involve e.g. silicon-based storage media and utilize model-based compression technology. Despite these advances in technology, the navigation solutions presented in this thesis will still be applicable provided that images or image parts are still coded as recoverable data fragments, since they separate the involved navigation processing over the recording and playback phase, which enables to balance the involved signal processing. Furthermore, the solutions presented in this thesis re-use normal-play compressed pictures, either intra-coded or predictive-coded, which are conceptual video compression approaches, which will be also available, most probably in a more advanced form, in future video compression algorithms.

It can be observed that the consumer storage systems developed in the past decade are equipped with conventional video navigation features, while more advanced navigation features are still absent. This is likely caused by

two reasons. The first is based on the cost constraints associated with video navigation, which are traditionally kept at a low level, while more advanced navigation features will exceed the cost budget for conventional video navigation. A second reason is the absence for a strong demand on advanced video navigation features in video storage systems. Although the storage capacity of consumer recording devices has grown over time, the conventional video navigation feature is still considered sufficient for typical navigation needs.

In the future, with the increased video services offered in the cloud, all video-related processing will be shifting away from the end user towards centralized storage systems. In order to efficiently navigate through this enormous amount of video information, there will be a larger demand for advanced video navigation solutions. It is therefore expected that the work conducted in the field of intra- and interprogram video navigation will emerge via new cloud-based video distribution solutions.

Mobile IP-based television

Future battery-powered mobile-based communication systems will most probably employ IP-based data protected by an additional forward error correction layer, to improve the reception performance. For such future systems, two system aspects described in this thesis remain suitable. First, in order to improve the reception robustness, the usage of error and erasure decoding, whereby the erasure flags are derived on small-sized data fragments. Second, the avoidance of IP datagram duplication, which reduces data bandwidth and lowers the energy consumption.

Artifact-location detection

Coding artifacts are inevitable and inherently associated with lossy video compression, making artifact detection a basic function as part of the video enhancement processing. Coding artifacts may vary for different compression schemes, which result in a different enhancement processing. It is expected that ringing, as it finds its origin in the Gibbs phenomenon, will remain a basic artifact inevitably associated with the huge legacy of block-based video compression standards (even emerging ones) and requires locally-adaptive filtering in order to attenuate this visible distortion. Furthermore, depending on the compression scheme, local image enhancement may also involve locally-adaptive sharpening. This would serve the reconstruction of edges, which suffered from high-frequency detail removal due to compression. The artifact-detection approaches presented in this thesis enable location detection of regions, which exhibit blurring and ringing artifacts caused by the attenuation and quantization of the high-frequency components. These deteriorations not only occur in all MPEG standards so far, but it is also the case for low-rate JPEG2000-compressed images based on wavelet coding.

Appendices

MPEG-2 Adaptation field

The MPEG-2 transport stream packet can be equipped with an adaptation field. This field is used for various multiplexing options, such as stuffing at TS level, indicated by *adaptation_field_length* and the transmission of the Program Clock Reference (PCR), indicated by *PCR_flag*. For the remaining fields, see [36]. The PCR is transmitted as a 33-bit *program_clock_reference_base* value, see equation A.1a and a 9-bit *program_clock_reference_extension* value, see equation A.1b. Hereby the *system_clock_reference* is a 27MHz clock.

$$PCR_base(i) = ((system_clock_reference \times tp(i)) \text{ DIV } 300) \text{ mod } 2^{33} \quad (\text{A.1a})$$

$$PCR_ext(i) = ((system_clock_reference \times td(i)) \text{ DIV } 1) \text{ mod } 300 \quad (\text{A.1b})$$

The PCR indicates the intended time of arrival of the byte containing the last bit of the *program_clock_reference_base* at the input of the system target decoder.

MPEG-2 Timestamps

In a Packetized Elementary Stream (PES), timestamps indicated by the *PTS_**DTS_flags* indicates the presence of *Decoding Time Stamp* (DTS) and presentation time *Presentation Time Stamp* (PTS), which are calculated according to Eqn. B.1a—B.1b, whereby the *system_clock_reference* is a 27MHz clock.

$$PTS(j) = ((system_clock_reference \times tp(j)) \text{ DIV } 300) \text{ mod } 2^{33} \quad (\text{B.1a})$$

$$DTS(j) = ((system_clock_reference \times td(j)) \text{ DIV } 300) \text{ mod } 2^{33} \quad (\text{B.1b})$$

Hereby $td(j)$ denotes the decoding time in the system target decoder of an access unit j . Similarly, $tp(j)$ denotes the presentation time in the system target decoder of an access unit j . Figure B.1 visualizes a video decoding example of the DTS and PTS values in relation to the PCR for a 25 Hz based television system. In Fig. B.1, $DTS(j)$ indicates the moment when the decoding process can

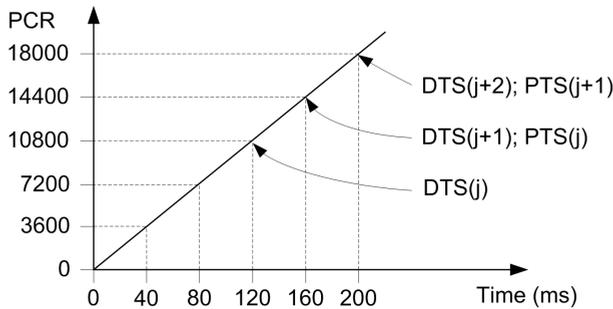


Figure B.1 — Program clock reference example.

start, the time prior to $DTS(j)$ is required to fill the video Elementary Stream (ES) decoder buffer, such that it contains at least a complete access unit. The decoded picture associated to $DTS(j)$ is presented one frame period later, indicated by $PTS(j)$, whereby the next compressed picture is decoded, as indicated

by $DTS(j + 1)$. Such a situation applies for MPEG-2 Simple Profile compressed video, which consists of only I-type and P-type pictures.

The *DSM_trick_mode_flag* indicates the presence of information regarding trick-play playback-modes such as fast-search modes and slow-search modes or freeze-frame display. However, although the usage of trick modes associated to the *DSM_trick_mode_flag* is recommended for decoding systems equipped with a digital interface, this is not demanded [37]. As a result, the support for these trick mode facilities is not reliable and other methods are required, circumventing the need for *DSM_trick_mode_flag* and its associated playback, while obtaining similar or equal trick-play playback [36].

MPEG-2 Section Syntax

A section is a syntactic structure that can be regarded as a container and defined by ISO/IEC 13818-1 [36]. The section syntax structure is deployed by MPEG-2 Program Specific Information (PSI) tables [36] and Service Information (SI) tables [130], defined by the DVB Project. The SI sections syntactic structures comply to the private section syntax defined by ISO/IEC 13818-1 [36]. Sections are carried by TS packets with either predetermined Packet Identifier (PIDs) or by user selectable PIDs and offer error detection by means of a Cyclic Redundancy Check (CRC) [36]. A section starts with a *Table_id*, which serves as an identification number. For the situation that the *Table_id* is present in that particular

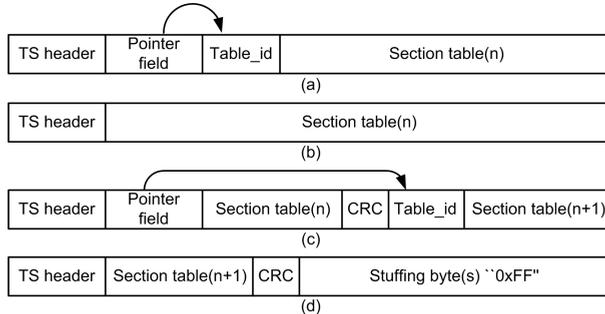


Figure C.1 — TS packet examples containing section data. (a) TS packet containing first section byte. (b) TS packet with only section payload. (c) TS packet with last part of section and start of new section. (d) TS packet with last part of section followed by stuffing bytes.

TS packet, the TS header is followed by a *pointerfield*, which on its turn may be preceded by an *adaptationfield*. This *pointerfield* points to the position where the first byte, *Table_id* of a section starts. When a section finishes, the next byte position, if available, contains either the *Table_id* of the next section or "0xFF".

If “0xFF”, then this indicates stuffing data, which will fill up the remaining TS payload byte positions of a particular TS packet, see Fig. C.1. Figure C.1 indicates some typical TS packet examples containing section payload, whereby n is a linear index indicating successive sections. The first eight bytes of a section form a generic part, followed by section dependable part, see Fig. C.3 and Fig. C.2. Figure C.3 and Fig. C.2 show two sections which are used by Digital Video Broadcasting Handheld (DVB-H) for the transmission of IP data and if available, the Reed-Solomon parity data. Beside the transmission of SI/PSI table information, typically deploying a data carousel-based transmission model, sections are also used for multi-protocol encapsulation, like deployed in DVB-H. Hereby an IP datagram, see Fig. C.3 is encapsulated into a Multi Protocol Encapsulation (MPE) section and the RS parity data, see Fig. C.2 is encapsulated into a Multi Protocol Encapsulation Forward Error Correction (MPEFEC) section. The MPE section is based on the DSM-CC section [96] and modified according to [97], while the MPEFEC sections are based on DSMCC_section Type “User private” [96], [97].

Syntax	Number of bits
MPE-FEC_section() {	
table_id	8
section_syntax_indicator	1
private_indicator	1
reserved	2
section_length	12
padding_columns	8
reserved_for_future_use	8
reserved	2
reserved_for_future_use	5
current_next_indicator	1
section_number	8
last_section_number	8
real_time_parameters()	
for (j=0; j<N1; j++) {	
rs_data_byte	8
}	
CRC_32	32
}	

Figure C.2 — *DVB-H Multi-Protocol Encapsulation Forward Error Correction (MPE-FEC) section.*

Syntax	Number of bits
datagram_section() {	
table_id	8
section_syntax_indicator	1
private_indicator	1
reserved	2
section_length	12
MAC_address_6	8
MAC_address_5	8
reserved	2
payload_scrambling_control	2
address_scrambling_control	2
LLC_SNAP_flag	1
current_next_indicator	1
section_number	8
last_section_number	8
MAC_address_4	8
MAC_address_3	8
MAC_address_2	8
MAC_address_1	8
if (LLC_SNAP_FLAG == '1') {	
LLC_SNAP()	
} else {	
for (j=0; j<N1; j++) {	
IP_datagram_data_byte	8
}	
}	
if (section_number == last_section_number) {	
for (j=0; j<N2; j++) {	
stuffing_byte	8
}	
}	
if (section_syntax_indicator == '0') {	
checksum	32
} else {	
CRC_32	32
}	
}	

Figure C.3 — DVB-H Multi-Protocol Encapsulation (MPE) section based on a modified DSM-CC section.

Characteristic Point Information

In order to optimize the involved signal processing during video navigation, Characteristic Point Information (CPI) is derived involving signal processing during recording, which is stored as metadata. The involved signal processing determines features of the MPEG-2-compressed normal-play video sequence that are not available as MPEG-2 syntax elements. Moreover, to facilitate efficient fast-search video navigation the start location of I-pictures is determined and stored as metadata. Algorithm 17 indicates the involved CPI signal processing.

Algorithm 17 Video navigation record processing

Initialize:

$GOPentry = 1, TSpacketcnt = 0, N = 0, Bpictcnt = 0, fr =$
television frame rate

while not end of recording **do** ▷ record processing on normal-play video

 ▷ The recorded TS is demultiplexed to access video elementary stream

 demultiplex TS packet ▷ MPEG-2 demux current TS packet

 ▷ Byte level parsing of video elementary stream

while *nextbytes* $\neq 0x001$ **do** ▷ search header prefix

nextbytes = *nextbytes* $\ll 8 + nextbyte$

end while

 read *start code*

return *start code*

 ▷ Process each individual picture

if *start code* == *picture header* **then**

$N = N + 1$

if *picture* == *Ipicture* **then**

$M = N / (N - Bpictcnt)$ ▷ Calculate M for previous GOP

$metadata[GOPentry - 1].M = M$ ▷ Store M for previous GOP

$metadata[GOPentry - 1].N = N$ ▷ Store N for previous GOP

$metadata[GOPentry].address = TScnt * 188$ ▷ I-pic. start address

 determine *Ipicture size* ▷ Determine current I-picture size

$nr_rep_pict = (\lceil fr * Ipicture\ size / maxbitrate \rceil - 1)$

$metadata[GOPentry].transmission = nr_rep_pict$

$GOPentry = ++$

$N = 0$

$Bpictcnt = 0$

$metadata[GOPentry].pict = subpict$ ▷ Sub-picture

$metadata[GOPentry].opict = subpictOSD$ ▷ OSD Sub-picture

$nr_rep_pict = (\lceil fr * mosaic_screen\ size / maxbitrate \rceil - 1)$

$metadata[GOPentry].mosaic_transmission = nr_rep_pict$

end if

if *picture* == *Bpicture* **then**

$Bpictcnt = ++$

 ▷ count all B-pictures per GOP

end if

end if

$TScnt = TScnt + 1$

 ▷ count all normal-play TS packets

end while

Artifact-location detection on full-HD upscaled video

Table E.1 — *Detection score performance in percentages of noise-pattern pixel detection in the spatial domain and frequency domain for upscaled HD resolution video. The kernel and block sizes are indicated in pixels.*

	Spatial-domain kernel and block size			Frequency-domain kernel and block size
	11 × 11	19 × 19	21 × 21	20 × 20
kernel	11 × 11	19 × 19	21 × 21	20 × 20
block size	3 × 3	mixed	5 × 5	4 × 4
JPEG Q=25				
True detected	8%	46%	40%	54%
Expansion detected	21%	40%	44%	33%
Total	29%	86%	84%	87%
JPEG Q=50				
True detected	14%	54%	49%	57%
Expansion detected	32%	32%	40%	29%
Total	46%	86%	89%	86%

Publication List

The following conference papers, journal papers and book chapters have been published based on the research presented in this thesis.

- [10] J.P. van Gassel, D.P. Kelly, O. Eerenberg and P.H.N. de With. "MPEG-2 Compliant Trick play over a Digital Interface". In: *Proc. Int. Conf. Consum. Electron.* June 2002, pp. 170–171. ISBN: 0-7803-7300-6.
- [11] O. Eerenberg and P.H.N. de With. "System Requirements and Considerations for Visual Table of Contents in Personal Video Recording". In: *Proc. Int. Conf. Consum. Electron.* June 2003, pp. 24–25. ISBN: 0-7803-7721-4.
- [12] O. Eerenberg and P.H.N. de With. "MPEG-2 Compliant Trick play over a Digital Interface". In: *IEEE Trans. Consum. Electron.* 51.3 (Aug. 2005), pp. 958–966.
- [13] O. Eerenberg and P.H.N. de With. "System Requirements and Considerations for Visual Table of Contents in PVR". In: *IEEE Trans. Consum. Electron.* 54.3 (Aug. 2008), pp. 1206–1214.
- [14] O. Eerenberg, and P.H.N. de With. "Digital Video". In: Intech, Vukovar Croatia, Feb. 2010. Chap. 15. ISBN: 978-9537619701.
- [16] O. Eerenberg, R.M. Aarts and P.H.N. de With. "System Design of Advanced Video Navigation Reinforced with Audible Sound in Personal Video Recording". In: *Proc. Int. Conf. Consum. Electron.* Jan. 2008, pp. 1–2. ISBN: 1-4244-1458-1.
- [22] O. Eerenberg, A. Koppelaar, A. Stuivenwold and P.H.N. de With. "IP-recovery in the DVB-H Link Layer for TV on Mobile". In: *Proc. Int. Conf. Consum. Electron.* Jan. 2006, pp. 411–412. ISBN: 0-7803-9459-3.
- [23] A.G.C. Koppelaar, O. Eerenberg, L.M.G.M. Tolhuizen and V. Aue. "Restoration of IP-datagrams in the DVB-H link-layer for TV on mobile". In: *Proc. Int. Conf. Consum. Electron.* Jan. 2006, pp. 409–410. ISBN: 0-7803-9459-3.

- [24] O. Eerenberg, P. Wendrich, E.H.W. van Orsouw and P.H.N. de With. "Efficient Validation/Verification of a Robust DVB-H Link Layer". In: *Proc. Int. Conf. Consum. Electron.* Jan. 2007, pp. 15–16. ISBN: 1-4244-0762-1.
- [25] O. Eerenberg, A. Koppelaar, A. Stuivenwold and P.H.N. de With. "IP-Recovery in the DVB-H Link Layer for TV on Mobile". In: *IEEE Trans. Consum. Electron.* 57.2 (May 2011), pp. 339–347.
- [26] O. Eerenberg, P. Wendrich, E.H.W. van Orsouw and P.H.N. de With. "Efficient Validation/Verification of a Robust DVB-H Link Layer". In: *IEEE Trans. Consum. Electron.* 57.5 (Nov. 2011), pp. 1679–1687.
- [27] O. Eerenberg, A. Koppelaar and P.H.N. de With. "Mobile Multimedia Broadcasting Standards: Technology and Practice". In: Springer, New York USA, Nov. 2008. Chap. 8. ISBN: 978-0387782621.
- [28] O. Eerenberg, J. Kettenis and P.H.N. de With. "Block-Based Detection Systems for Visual Artifact Location". In: *Proc. Int. Conf. Consum. Electron.* Jan. 2013, pp. 116–117. ISBN: 978-1-4673-1362-9.
- [29] O. Eerenberg, J. Kettenis and P.H.N. de With. "Block-based detection systems for visual artifact location". In: *IEEE Trans. Consum. Electron.* 59.2 (May 2013), pp. 376–384.
- [131] O. Eerenberg, Y. Gao, E. Trauschke and P.H.N. de With. "Pattern-Based De-Correlation for Visual-Lossless Video Compression for Wireless Display Applications". In: *Proc. Int. Conf. Consum. Electron.* Jan. 2010, pp. – 229–230. ISBN: 978-1-4244-4315-4.

Complete Bibliography

- [1] DVB Project. <http://www.dvb.org>.
- [2] A/53: ATSC Digital Television Standard, Parts 1 - 6, 2007. Jan. 2004.
- [3] CEI/IEC-61834: "Helical-scan digital video cassette recording system using 6,35 mm magnetic tape for consumer use (525-60, 625-50, 1125-60 and 1250-50 systems. 1998.
- [4] ISO/IEC 11172-2, Information technology - Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s - Part 2: Video. Aug. 1993.
- [5] ITU-T Rec. H.262 and ISO/IEC 13818-2 (MPEG2), Generic Coding of Moving Pictures and Associated Audio Information Part 2: Video. Nov. 1994.
- [6] O. Eerenberg and A.M.A Rijckaert. Patent Application – Digital information recording apparatus using tracks on record carrier uses first and second digital information signals to create first recording signal and first trick play signal, suitable for recording in tracks of record carrier. US6400888B1, June 2002.
- [7] O. Eerenberg, D.P. Kelly and J.P. Gassel. Patent Application – Encoded video signal for TV based trick play, has empty interlaced elimination picture displayed immediately following original picture. US20020167607A1, Nov. 2002.
- [8] O. Eerenberg, E. Gijsbers and E. Brandsma. Patent Application – Compressed video signal sub-pictures manipulating method, involves manipulating sub-picture of compressed video signal by manipulating association of control data with compressed picture blocks related to sub-picture. US20060050790A1, Mar. 2006.
- [9] W.H.A. Bruls, O. Eerenberg and A.M.A. Rijckaert. Patent Application – Trick play signal generation method for digital video recorder. US20030231-863A1, Dec. 2003.

- [10] J.P. van Gassel, D.P. Kelly, O. Eerenberg and P.H.N. de With. "MPEG-2 Compliant Trick play over a Digital Interface". In: *Proc. Int. Conf. Consum. Electron.* June 2002, pp. 170–171. ISBN: 0-7803-7300-6.
- [11] O. Eerenberg and P.H.N. de With. "System Requirements and Considerations for Visual Table of Contents in Personal Video Recording". In: *Proc. Int. Conf. Consum. Electron.* June 2003, pp. 24–25. ISBN: 0-7803-7721-4.
- [12] O. Eerenberg and P.H.N. de With. "MPEG-2 Compliant Trick play over a Digital Interface". In: *IEEE Trans. Consum. Electron.* 51.3 (Aug. 2005), pp. 958–966.
- [13] O. Eerenberg and P.H.N. de With. "System Requirements and Considerations for Visual Table of Contents in PVR". In: *IEEE Trans. Consum. Electron.* 54.3 (Aug. 2008), pp. 1206–1214.
- [14] O. Eerenberg, and P.H.N. de With. "Digital Video". In: Intech, Vukovar Croatia, Feb. 2010. Chap. 15. ISBN: 978-9537619701.
- [15] R. Aarts and G. Bloemen. *Patent Application – Method and apparatus for providing a video signal.* US20070035666A1, Oct. 2003.
- [16] O. Eerenberg, R.M. Aarts and P.H.N. de With. "System Design of Advanced Video Navigation Reinforced with Audible Sound in Personal Video Recording". In: *Proc. Int. Conf. Consum. Electron.* Jan. 2008, pp. 1–2. ISBN: 1-4244-1458-1.
- [17] O. Eerenberg, R.M. Aarts and P.H.N. de With. "PVR system design of advanced video navigation reinforced with audible sound". In: *IEEE Trans. Consum. Electron.* 60.4 (Nov. 2014), pp. 681–689.
- [18] A.G.C. Koppelaar, O. Eerenberg, and M.G. Verhoeven. *Patent Application – Packet reconstruction apparatus for digital video broadcast system for handheld, stores and ignores fragment and its length in column along with its associated length variable when fragment is in packet with un-correctable error.* US20080209477A1, Aug. 2008.
- [19] A.G.C. Koppelaar, L.M.G.M. Tolhuizen and O. Eerenberg. *Patent Application – Packet reconstruction apparatus for digital video broadcast system for handheld, stores and ignores fragment and its length in column along with its associated length variable when fragment is in packet with un-correctable error.* US20080282310A1, Nov. 2008.
- [20] A.G.C. Koppelaar, O. Eerenberg and A.M. Stuivenwold. *Patent Application – Mobile device for receiving bursts in communications network requests end of reception of error correction data when amount of correctly received data is less than necessary data amount.* US20090006926A1, Jan. 2009.

- [21] O. Eerenberg, A.G.C. Koppelaar and A.M. Stuivenwold. *Patent Application – Device e.g. mobile device used in transmission system using digital video broadcasting standards, has memory unit overwriting forward error correction data of one burst with data of other burst*. US20080263428A1, Oct. 2008.
- [22] O. Eerenberg, A. Koppelaar, A. Stuivenwold and P.H.N. de With. “IP-recovery in the DVB-H Link Layer for TV on Mobile”. In: *Proc. Int. Conf. Consum. Electron.* Jan. 2006, pp. 411–412. ISBN: 0-7803-9459-3.
- [23] A.G.C. Koppelaar, O. Eerenberg, L.M.G.M. Tolhuizen and V. Aue. “Restoration of IP-datagrams in the DVB-H link-layer for TV on mobile”. In: *Proc. Int. Conf. Consum. Electron.* Jan. 2006, pp. 409–410. ISBN: 0-7803-9459-3.
- [24] O. Eerenberg, P. Wendrich, E.H.W. van Orsouw and P.H.N. de With. “Efficient Validation/Verification of a Robust DVB-H Link Layer”. In: *Proc. Int. Conf. Consum. Electron.* Jan. 2007, pp. 15–16. ISBN: 1-4244-0762-1.
- [25] O. Eerenberg, A. Koppelaar, A. Stuivenwold and P.H.N. de With. “IP-Recovery in the DVB-H Link Layer for TV on Mobile”. In: *IEEE Trans. Consum. Electron.* 57.2 (May 2011), pp. 339–347.
- [26] O. Eerenberg, P. Wendrich, E.H.W. van Orsouw and P.H.N. de With. “Efficient Validation/Verification of a Robust DVB-H Link Layer”. In: *IEEE Trans. Consum. Electron.* 57.5 (Nov. 2011), pp. 1679–1687.
- [27] O. Eerenberg, A. Koppelaar and P.H.N. de With. “Mobile Multimedia Broadcasting Standards: Technology and Practice”. In: Springer, New York USA, Nov. 2008. Chap. 8. ISBN: 978-0387782621.
- [28] O. Eerenberg, J. Kettenis and P.H.N. de With. “Block-Based Detection Systems for Visual Artifact Location”. In: *Proc. Int. Conf. Consum. Electron.* Jan. 2013, pp. 116–117. ISBN: 978-1-4673-1362-9.
- [29] O. Eerenberg, J. Kettenis and P.H.N. de With. “Block-based detection systems for visual artifact location”. In: *IEEE Trans. Consum. Electron.* 59.2 (May 2013), pp. 376–384.
- [30] *ISO/IEC 11172-1, Information technology - Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s - Part 1: Systems*. Aug. 1993.
- [31] *ISO/IEC 11172-3, Information technology - Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s - Part 3: Audio*. Aug. 1993.
- [32] *EN 300 468: Digital Video Broadcasting (DVB); Specification for Service Information (SI) in DVB systems*. Feb. 1998.
- [33] *ETSI EN 302 304: “Digital Video Broadcasting (DVB); Transmission System for Handheld Terminals (DVB-H)”*. Nov. 2004.

- [34] *Draft ETSI TR 102 377 v1.1.1: Digital Video Broadcasting (DVB); DVB-H Implementation Guidelines*. Feb. 2005.
- [35] M. Yuen and H.R. Wu. "A survey of hybrid MC/DPCM/DCT video coding distortions". In: *Signal process. special issue on image and video quality metrics* 70.3 (Nov. 1998), pp. 247–278.
- [36] *ITU-T Rec. H.262 and ISO/IEC 13818-1 (MPEG2), Generic Coding of Moving Pictures and Associated Audio Information Part 1: Systems*. Nov. 1994.
- [37] *Digital Video Broadcasting (DVB); Specification for the use of Video and Audio Coding in Broadcasting Applications based on the MPEG-2 Transport Stream*. Nov. 2012.
- [38] S. Baggen. "Digital consumer electronics handbook". In: McGraw-Hill, Hightstown, NJ, USA, May 1997. Chap. 4. ISBN: 0-07-034143-5.
- [39] *ITU-T Rec. H.262 and ISO/IEC 13818-3 (MPEG2), Generic Coding of Moving Pictures and Associated Audio Information Part 3: Audio*. Nov. 1994.
- [40] *ITU-T Rec. H.264 and ISO/IEC 14496-10 (MPEG4-AVC), Advanced Video Coding for Generic Audiovisual Services, v4*. July 2005.
- [41] A. Luthra, G. J. Sullivan, and T. Wiegand. "Introduction to the Special Issue on the H.264/AVC Video Coding Standard". In: *IEEE Trans. Circuits Syst. Video Technol.* 13.7 (July 2003), pp. 557–559.
- [42] C. Fenimore, J. Libert and P. Roitman. "Mosquito noise in MPEG compressed video: test patterns and metrics". In: vol. 3959. June 2000, pp. 604–612. ISBN: 978-0-8194-3577-4.
- [43] S.J.P. Westen, R.L. Lagendijk, and J. Biemond. "Adaptive spatial noise shaping for dct based image compression". In: *Proc. Int. Conf. Acoust., Speech, Signal Process.* May 1996, 21242127. ISBN: 0-7806-3193-1.
- [44] C. Mantel, P. Ladret and T. Kunlin. "Temporal mosquito noise corrector". In: *Int. Workshop Quality of Multimedia Experience*. July 2009, pp. 29–31. ISBN: 978-1-4244-4370-3.
- [45] D.T. Vo, T.Q. Nguyen, S. Yea and A. Vetro. "Coding artifacts reduction using edge map guided adaptive and fuzzy filtering". In: *IEEE Trans. Image Process.* 18.6 (June 2009), pp. 1166–1178.
- [46] *ETSI TS 182 027 V2.0.0, TISPAN; IPTV Architecture; IPTV functions supported by the IMS subsystem*. Feb. 2008.
- [47] S.Y. Lim, J.H. Choi, J.M. Seok and H.K. Lee. "Advanced PVR Architecture with Segment-based Time-Shift". In: *Proc. Int. Conf. Consum. Electron.* Jan. 2007, pp. 1–2. ISBN: 1-4244-0763-X.
- [48] J.H. Lee, G.G. Lee and W.Y. Kim. "Automatic video summarizing tool using MPEG-7 descriptors for personal video recorder". In: *Proc. Int. Conf. Consum. Electron.* June 2003, pp. 742–749. ISBN: 0-7803-7721-4.

- [49] F.E. Sandnes, Y.P. Huang and Y.M. Huang. "Browsing of Large Video Collections on Personal Video Recorders with Remote Control Navigation Keys". In: *Proc. Int. Conf. Future Generation Commun. Networking*. Dec. 2008, pp. 422–427. ISBN: 978-0-7695-3431-2.
- [50] S. Suh J.R. Kim and S. Sul. "Design and implementation of an enhanced personal video recorder for DTV". In: *IEEE Trans. Consum. Electron.* 47.4 (Nov. 2001), pp. 864–869. ISSN: 0098-3063.
- [51] K.W. Tindell. *A digital video recorder system connectable to devices running a web browser*. WO2012032174A1, Mar. 2012.
- [52] B. Here, Y. Poupet and L. Vantalou. *Method and Apparatus for Trick Mode Operation of a Personal Video recorder Over a Network*. EP-2383740, Apr. 2011.
- [53] E. Mikoczy, R. Kadlic and P. Podhradsky. "Advance PVR Applications in IMS Based IPTV Environment". In: *Proc. Int. Conf. Syst., Signals and Image Process.* June 2009, pp. 1–4. ISBN: 978-1-4244-4530-1.
- [54] F. Callaly and P.M. Corcoran. "Architecture of a PVR appliance with 'long-tail' Internet-TV capabilities". In: *IEEE Trans. Consum. Electron.* 52.2 (May 2006), pp. 454–459.
- [55] E.S. Kim S.W. Jung and D.H. Lee. "Design and implementation of an enhanced personal video recorder for DTV". In: *IEEE Trans. Consum. Electron.* 47.4 (Nov. 2001), pp. 864–869. ISSN: 0098-3063.
- [56] M.F. Demeyer. *Patent Application – Apparatus and method for automated video editing*. US8290334B2, Oct. 2012.
- [57] V. Iverson, G. Martz, J. Mcveigh, R. Rao, K. Salzberg, S. Sirivara and D. Wagner. *Transcoding media content from a personal video re-corder for a portable device*. WO2003107672A1, Mar. 2005.
- [58] H. Yuen. *Patent Application – Personal Video Recorder with High-Capacity Archive*. US-20020186957A1, Apr. 2002.
- [59] A. Divakaran and R. Cabasson. "Content-based browsing system for personal video recorders". In: *Proc. Int. Conf. Consum. Electron.* June 2002, pp. 114–115. ISBN: 0-7803-7300-6.
- [60] A. Divakaran. *Multimedia Content Analysis*. Springer Verlag, 2009. ISBN: 978-0-387-76567-9.
- [61] M.D. Ellis. *Recommendation of Media Content on a Media System*. WO-2003098932A1, Apr. 2003.
- [62] M.D. Ellis. *Systems and Methods for Interactive Program Guides with Personal Video Recording Features*. WO2002069636A1, Feb. 2002.

- [63] B. Engelbert, M. Blanken, R. Kruthoff and K. Morisse. "A user supporting personal video recorder by implementing a generic Bayesian classifier based recommendation system". In: *Proc. Int. Conf. Pervasive Comput. Commun. Workshops*. Mar. 2011, pp. 567–571. ISBN: 978-1-61284-938-6.
- [64] S.Y. Lim, J.H. Choi, J.M. Seok and H.K. Lee. "Advanced PVR Architecture with Segment-based Time-Shift". In: *Proc. Int. Conf. Consum. Electron.* Jan. 2007, pp. 1–2. ISBN: 1-4244-0762-1.
- [65] J. Park. *Patent Application – Method of searching scenes recorded in PVR and television receiver using the same*. US20070040936A1, Feb. 2007.
- [66] H. Kim, J. Kim, M. Rostoker, Y.S. Seong and S. Sull. *Patent Application – Techniques for navigating multiple video streams*. US20060064716A1, Mar. 2006.
- [67] G.G. Lee J.H. Lee and W.Y. Kim. "Automatic video summarizing tool using MPEG-7 descriptors for personal video recorder". In: *IEEE Trans. Consum. Electron.* 49.3 (Aug. 2003), pp. 742–749. ISSN: 0098-3063.
- [68] Y. Ruiy, Z. Xiongy, R. Radhakrishnanz, A. Divakaranz and T.S. Huangy. *A Unified Framework for Video Summarization Browsing and Retrieval*. Academic Press, 2005. ISBN: 978-0-12-369387-7.
- [69] S.H. Lee. *Patent Application – Personal Video Recorder and method for operating the same*. US20030128969A1, Jan. 2003.
- [70] R.R. Dunton, L.A. Booth and B. Hakimi. *Patent Application – Image-Keyed Index for Video Program Stored in Personal Video Recorder*. US20060110-128A1, May 2006.
- [71] W.S. Herz. *Patent Application – Video Navigation System and Method*. US-20090074377A1, Mar. 2009.
- [72] A.F. Elcock and J. Kamienicki. *Patent Application – Navigation Recorded Video Using Closed Captioning*. US20070154171A1, July 2007.
- [73] A.F. Elcock and J. Kamienicki. *Patent Application –Navigation Recorded Video Using Captioning, Dialogue and Sound*. US20070154176A1, July 2007.
- [74] S.J.G Min. *Motional video browsing data structure and browsing method therefor*. EP1006461, June 2000.
- [75] E. Belinsky, E. Shahar. *Patent Application – Method and System for Navigation of Audio and Video Files*. US20100141655A1, June 2010.
- [76] J.R. Kim, S. Suh and S. Sull. "Fast scene change detection for personal video recorder". In: *IEEE Trans. Consum. Electron.* 49.3 (Aug. 2003), pp. 683–688.
- [77] J.G. Jeong, C.K. Jung, S. Moon and S.K. Kim. *Patent Application – Method, Medium and System Generating Navigation Information of Input Video*. US-20070291986A1, Dec. 2007.

- [78] M. Bober, S. Paschalakis. *Method and Apparatus for Video Navigation*. WO-2007028991A1, Mar. 2007.
- [79] C.R. Johnson. *Patent Application – User-specific time values for time-based navigation functions of video recorder systems*. US6865336B2, Mar. 2005.
- [80] S. Mo, C.J. Ochoa, V. Szilagy and E. Smith. *Method and Apparatus for Keyword-Based, Non-Linear Navigation of Video Streams and Other Content*. WO2013039473A1, Sept. 2011.
- [81] D.G. Cronin and R.A. Morris. *Patent Application – Thumbnail Navigation Bar for Video*. US20090172543A1, July 2009.
- [82] M. Teicher, E. Lev and N. Cohen. *Patent Application – Browsing System, Method and Apparatus for Video Motion Pictures*. WO2000018120, Mar. 2000.
- [83] W.S. Herz. *Video Perspective Navigation System and Method*. US20090160-933A1, June 2009.
- [84] *International Standard ISO/IEC 14496-3, Information technology —Coding of audio-visual objects —Part 3: Audio*. Dec. 2001.
- [85] M. Fujita et al. “Newly developed D-VHS digital tape recording system for the multimedia era”. In: *IEEE Trans. Consum. Electron.* 42.3 (Aug. 1996), pp. 617–622.
- [86] C. Buma et al. “DVD+RW: 2-Way Compatibility for Video and Data Applications”. In: *Proc. Int. Conf. Consum. Electron.* Jan. 2000, pp. 88–89.
- [87] D.P. Kelly, W. van Gestel, T. Hamada, M. Kato and K. Nakamura. “Blu-ray disc - a versatile format for recording high definition video”. In: *Proc. Int. Conf. Consum. Electron.* June 2003, pp. 72–73. ISBN: 0-7803-7721-4.
- [88] S.O. Mietens, P.H.N. de With and C. Hentschel. “New DCT Computation Technique based on Scalable Resources”. In: *IEEE Workshop Signal Process. Syst.* (Sept. 2001), pp. 285–296.
- [89] *System Description Blu-ray Disc Read-Only Format, Part 3: Audio Visual Basic Specifications (3-1 Core Specifications) Version 2.01 DRAFT2*. July 2006.
- [90] D. L. McLaren. *HDTV trick-play stream derivation for VCR*. EP0787403B1, Sept. 1995.
- [91] C. Yingwei, Z. Zhun, L. Tse-Hua, S. Peng and K. van Zon. “Regulated complexity scalable MPEG-2 video decoding for media processors”. In: *IEEE Trans. Circuits Syst. Video Technol.* 12.8 (Aug. 2002), pp. 678–687.
- [92] *ISO/IEC 13818-5: Information Technology - Generic Coding of Moving Pictures and Associated Audio Recommendation - Software-Simulation*. Nov. 1994.
- [93] Eun-Seok Kim, Tae-Woong Um, and Seoung-Jun Oh. “A Fast Thumbnail Extraction Method in H.264/AVC Video Streams”. In: *IEEE Trans. Consum. Electron.* 55.3 (Aug. 2009), pp. 1424–1430.

- [94] J.R. Leonardi. "Multiple Time-Scales of Language Dynamics: An Example From Psycholinguistics". In: *Taylor & Francis, Ecological Psychology* (2010), pp. 269–285. ISSN: 1040-7413.
- [95] Chen Chen, Ping-Hao Wu and H. Chen. "Transform-Domain Intra Prediction for H.264". In: *IEEE ISCAS* (May 2005), pp. 1497–1500.
- [96] *ISO/IEC 13818-6: Information Technology - Generic Coding of Moving Pictures and Associated Audio Recommendation (Extensions for DSM-CC)*. Sept. 1998.
- [97] *ETSI EN 301 192: Digital Video Broadcasting (DVB); DVB specification for data broadcasting*. June 2004.
- [98] *ETSI TR 102 469: Digital Video Broadcasting (DVB); IP Datacast over DVB-H: Architecture*. May 2006.
- [99] J. Postel, "Internet Protocol - DARPA Internet Program Protocol Specification," RFC 791, USC/Information Sciences Institute. Sept. 1981.
- [100] S. Deering and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460. Dec. 1998.
- [101] *ISO/IEC 8802-2 Logical Link Control (LLC)*. 1998.
- [102] *ISO/IEC 8802-1a SubNetwork Attachment Point (SNAP) specification, IP-datacast baseline specification; Specification of interface LMT A80*. 2001.
- [103] *ETSI TS 102 471: Digital Video Broadcasting (DVB); IP Datacast over DVB-H: Electronic Service Guide (ESG)*. Mar. 2010.
- [104] *ISO/IEC 7498-1: "Information Technology - Open Systems Interconnection - Basic Reference Model: The Basic Model"*.
- [105] L.R. Siruvuri, P. Salama, and D.S. Kim. "Adaptive Error Resilience for Video Streaming". In: *Int. Journal of Digital Multimedia Broadcasting*. Vol. - 2009.
- [106] H. Joki, J. Paavola and V. Ipatov. "Analysis of Reed-Solomon Coding Combined with Cyclic Redundancy Check in DVB-H link layer". In: *Int. Symposium Wireless Commun. Systems*. Sept. 2005, pp. 313–317. ISBN: 0-7803-9206-X.
- [107] H. Joki and J. Paavola. "A Novel Algorithm for Decapsulation and Decoding of DVB-H Link Layer Forward Error Correction". In: *IEEE Int. Conf. Commun. (ICC)*. June 2006, pp. 5283–5288. ISBN: 1-4244-0355-3.
- [108] *ISO/IEC JTC 1, Coding of Audio-Visual Objects Part 2: Visual, ISO/IEC 14496-2 (MPEG4 Visual Version 2)*. Feb. 2000.
- [109] J. Hutchison. "Plasma display panels: the colorful history of an Illinois technology". In: *ECE alumni news, University of Illinois* 36.1 (Oct. 2002).

- [110] W. C. OMara. *Liquid crystal flat panel display: manufacturing science and technology*. Springer Verlag, 1993. ISBN: 978-0-442-01428-5.
- [111] Y.Juhn-Suk, J. Sang-Hoon, K. Yong-Chul, B. Seung-Chan, K. Jong-Moo, C. Nack-Bong, Y. Soo-Young, K. Chang-Dong, H. Yong-Kee and C. In-Jae. "Highly Flexible AM-OLED Display With Integrated Gate Driver Using Amorphous Silicon TFT on Ultrathin Metal Foil". In: *J. display technol.* 6.11 (Nov. 2010), pp. 565–570.
- [112] D. Tralic, J. Petrovic and S. Grgic. "JPEG image tampering detection using blocking artifacts". In: *Int. Conf. Systems, Signals, Image Process. (IWS-SIP)*. 2012, pp. 5–8.
- [113] H. Yao-Min, D. Leou and M. Cheng. "A Post Deblocking Filter for H.264 Video". In: *Proc. Int. Conf. Computer Commun. Networks (ICCCN)*. 2007, pp. 1137–1142.
- [114] A. Petrov, T. Kartalov and Z. Ivanovski. "Blocking effect reduction in low bitrate video on a mobile platform". In: *IEEE Int. Conf. Image Process. (ICIP)*. 2009, pp. 3937–3940.
- [115] I.O. Kirenko, S. Ling, R. Muijs and A. Nakonechny. "Enhancement of compressed video signals using a local blockiness metric". In: *IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*. 2008, pp. 1397–1400.
- [116] H. Liu and I. Heynderickx. "A perceptually relevant no-reference blockiness metric based on local image characteristics". In: *EURASIP J.Adv. Signal Process., vol. 2009*. Vol. 2009. 2009.
- [117] F.X. Coudoux, M.G. Gzalet, and P. Corlay. "A postprocessor for reducing temporal busyness in low-bit-rate video applications". In: *Signal Process. Image Commun.* 18 (Feb. 2003), 455463.
- [118] K. Jostschulte. "A new cascaded spatio-temporal noise reduction scheme for interlaced video". In: *Proc. Int. Conf. Image Process.* Oct. 1998, pp. 493–497. ISBN: 0-8186-8821-1.
- [119] P. and Y.T. Kim. *Patent Application – Method and apparatus for reducing mosquito noise in decoded video sequence*. US7657098B2, Feb. 2010.
- [120] P. and Y.T. Kim. *Patent Application – Video quality adaptive coding artifact reduction*. US7865035B2, Jan. 2011.
- [121] P. Hou, C. Lin, S. Kondo and C. Wu. "Reduction Of Image Coding Artifacts Using Spatial Structure Analysis". In: Feb. 2007, pp. 1–4.
- [122] L. Shao and I. Kirenko. "Content adaptive coding artifact reduction for decompressed video and images". In: *Proc. Int. Conf. Consum. Electron.* Jan. 2007, pp. 29–31. ISBN: 1-4244-0762-1.

- [123] H. Liu, N. Klomp and I. Heynderickx. "A Perceptually Relevant Approach to Ringing Region Detection". In: *IEEE Trans. Image Process.* 19.6 (2010), pp. 1414–1426.
- [124] A.V. Nasonov and A.S. Krylov. "Scale-space method of image ringing estimation". In: *Proc. 16th IEEE Int. Conf. Image Process. (ICIP)*. 2009, pp. – 2793–2796.
- [125] S. Chebbo, P. Durieux and B. Pesquet-Popescu. "Adaptive deringing and mosquito noise reducer". In: Jan. 2010.
- [126] H.S. Kong, Y. Nie, A. Vetro, H. Sun and K.E. Barner. "Coding Artifact Reduction Using Edge Map Guided Adaptive and Fuzzy Filter". In: Nov. 2004, pp. 1135–1138.
- [127] I. Kirenko. "Reduction of coding artifacts using chrominance and luminance spatial analysis". In: *Proc. Int. Conf. on Consum. Electron.* Jan. 2006, pp. 7–11. ISBN: 0-7803-9459-3.
- [128] Z. Qian, W. Wang and T. Qiao. "An Edge Detection Method in DCT Domain". In: *Procedia Engineering*. 2012, pp. 344–348.
- [129] I. Kirenko, S. Ling and A. Nakonechny. "Quality Enhancement of Compressed Video Signals". In: *Proc. Int. Conf. Consum. Electron.* Jan. 2008, 12. ISBN: 1-4244-1458-1.
- [130] *Digital Video Broadcasting (DVB); Specification for Service Information (SI) in DVB systems, DVB Document A38*. Jan. 2011.
- [131] O. Eerenberg, Y. Gao, E. Trauschke and P.H.N. de With. "Pattern-Based De-Correlation for Visual-Lossless Video Compression for Wireless Display Applications". In: *Proc. Int. Conf. Consum. Electron.* Jan. 2010, pp. – 229–230. ISBN: 978-1-4244-4315-4.
- [132] *ETSI TR 102 401 V1.1.1: Technical Report Digital Video Broadcasting (DVB); Transmission to Handheld Terminal (DVB-H); Validation Task Force Report*. May 2005.
- [133] *ISO/IEC 13818-4: Information Technology - Generic Coding of Moving Pictures and Associated Audio Recommendation H.222.0 (conformance)*. 1998.
- [134] H. Schulzrinne, S. Casner, R. Frederick and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications, RFC 1889". Jan. 1996.
- [135] J. Postel, "User Datagram Protocol", RFC 768. Aug. 1980.
- [136] *ITU-T Rec. H.263, Video Coding for Low Bit Rate Communication, v3*. Nov. 2000.
- [137] <http://iphome.hhi.de/suehring/tml/>.
- [138] *IEEE Standard 1012: IEEE Standard for Software Verification and Validation*. 1998.

- [139] R.J.J. Saeijs and F.J. Jorritsma. *Patent Application – Signal Processing On Information Files So As To Obtain Characteristic Point Information Sequences*. US6871007B1, Mar. 2005.
- [140] D. Marpe, T. Wiegand and S. Gordon. “H.264/MPEG4-AVC fidelity-range extensions: tools, profiles, performance, and application areas”. In: *IEEE Int. Conf. Image Process.* 1 (Sept. 2005), pp. I–593–6.
- [141] CEI/IEC-60774-1: “Helical-scan video tape cassette system using 12,65 mm (0.5 in) magnetic tape on type VHS, part 1: VHS and compact VHS video cassette system”. 1994.
- [142] CEI/IEC-60774-1: “Helical-scan video tape cassette system using 12,65 mm (0.5 in) magnetic tape on type VHS, part 5: D-VHS”. 2004.
- [143] P.H.N. de With and A.M.A. Rijckaert. “Design considerations of the video compression system of the new DV camcorder standard”. In: *IEEE Trans. Consum. Electron.* 43.4 (Nov. 1997), pp. 1160–1179.
- [144] H. Ting et al. “Trick play schemes for advanced television recording on digital VCR”. In: *IEEE Trans. Consum. Electron.* 41.4 (Nov. 1996), pp. 1159–1168.
- [145] H. Ting et al. “Fast scan technology for digital video tape recorders”. In: *IEEE Trans. Consum. Electron.* 39.3 (Aug. 1993), pp. 186–191.
- [146] IEC61883-4: “Digital Interface For Consumer Audio/Video Equipment Part 4: MPEG-TS data transmission”. 1998.
- [147] ISO/IEC 14496-2: “Information Technology Coding of audio-visual objects - Part 2: Visual”. 2001.
- [148] ISO/IEC 15938-1: “Information Technology - Multimedia Content Description Interface - Part 1: Systems”. 2001.
- [149] ISO/IEC 15938-3: “Information Technology Multimedia Content Description Interface - Part 3: Visual”. 2002.
- [150] I. Kirenko, R. Muijs and L. Shao. “Coding artifact reduction using non-reference block grid visibility measure”. In: *IEEE Int. Conf. Multimedia & Expo.* July 2006, pp. 469–472. ISBN: 1-4244-0367-7.
- [151] H. Hu and G. de Haan. “Simultaneous coding artifacts reduction and sharpness enhancement”. In: *Proc. Int. Conf. Consum. Electron.* Jan. 2007, pp. 1–2. ISBN: 1-4244-0762-1.
- [152] S. Choy, Y. Chan and W. Siu. “Reduction of Block-Transform Image Coding Artifacts by Using Local Statistics of Transform Coefficients”. In: *IEEE Signal Process. Lett.* 4.1 (Jan. 1997), pp. 5–7.
- [153] Y. Zhenghua Yu, W. Hong Ren, S. Winkler and C. Tao. “Vision-model-based impairment metric to evaluate blocking artifacts in digital video”. In: *Proc. of IEEE* 90 (1 2002), pp. 154–169.

- [154] Y. Nie, A. Vetro, S. Huifing and K.E. Barner. "Coding artifacts reduction using edge map guided adaptive and fuzzy filtering". In: June 2004, pp. 1135–1138. ISBN: 0-7803-7965-9.
- [155] L. Shao and I. Kirenko. "Coding Artifact Reduction Based on Local Entropy Analysis". In: *IEEE Trans. Consum. Electron.* 53.2 (May 2007), pp. – 691–696.
- [156] Y. Kato, T. Goto, S. Hirano and M. Sakurai. "Compression artifacts reduction for MPEG-2 video utilizing total variation regularization method". In: *Proc. Int. Conf. Consum. Electron.* (Jan. 2011), pp. 251–252.
- [157] I. Yankilevich. *Patent Application – Mosquito Noise Detection and Reduction*. US7949051B2, May 2011.
- [158] P.W. Chao and H. Y. Ou. *Patent Application – Image Process. Method and Device for Performing Mosquito Noise Reduction*. US2008085059A1, Apr. 2008.
- [159] J. Taylor. "DVD Demystified". In: McGraw-Hill, 2001. ISBN: 0-07-135026-8.
- [160] *IEC60774-5 Helical-scan video tape cassette system using 12,65 mm (0,5 in) magnetic tape on type VHS Part 5: D-VHS*. Apr. 2004.
- [161] *IEC62107 Super Video Compact Disc Disc-interchange system-specification International Standard*. July 2000.
- [162] *System Description Blu-ray Disc Read-Only Format, Part 3: Audio Visual Basic Specifications (3-1 Core Specifications) Version 2.5*. June 2011.
- [163] A.G. MacInnis. *System and Method for Personal Video Recording*. WO2002-043385A2, May 2002.
- [164] G. Aggarwal, A. Gopalakrishna Rao, M. Kellerman, D. Erickson, F. Demas, S. Bhatia and G. Hulmani. *Patent Application – Performing Personal Video Recording (PVR) Functions on Digital Video Streams*. US20030169815-A1, Sept. 2003.
- [165] *ETSI TS 181 016 V2.0.0, TISPAN; Service Layer Requirements to Integrate NGN Services and IPTV*. Nov. 2007.
- [166] J. Bae, D. Kim and H.I. Kang. "An Efficient Personal Video Recorder System". In: *Proc. Int. Conf. Intell. Comput. Technol. Autom.* May 2010, pp. 501–504. ISBN: 978-1-4244-7279-6.
- [167] J. Bae, D. Kim and H.I. Kang. "An Efficient Personal Video Recorder System". In: *Proc. Int. Conf. on Consum. Electron.* June 2002, pp. 114–115. ISBN: 0-7803-7300.
- [168] J.R. Kim, S. Suh and S. Sul. "Fast Scene Change Detection for Personal Video Recorder". In: *Proc. Int. Conf. Consum. Electron.* June 2003, pp. 742–749. ISBN: 0-7803-7721-4.

- [169] Su-Woon Jung and Dong-Ho Lee. "Modeling and analysis for optimal PVR implementation". In: *IEEE Trans. Consum. Electron.* 52.3 (2006), pp. 864–869. ISSN: 0098-3063.
- [170] J.R. Kim, S. Suh and S. Sul. "PVR a novel PVR scheme for content protection". In: *Proc. Int. Conf. Consum. Electron.* Jan. 2011, pp. 489–490. ISBN: 978-1-4244-8711-0.
- [171] H. Lee J. Son and H. Oh. "PVR a novel PVR scheme for content protection". In: *IEEE Trans. Consum. Electron.* 57.1 (Feb. 2011), pp. 173–177. ISSN: 0098-3063.
- [172] J.R. Kim, S. Suh and S. Sull. "Fast scene change detection for personal video recorder". In: *IEEE Trans. Consum. Electron.* 49.3 (Aug. 2003), pp. 683–688.
- [173] J. Bae, D. Kim and H.I. Kang. "An Efficient Personal Video Recorder System". In: *Proc. Int. Conf. Intell. Comput. Technol. Automation.* May 2010, pp. 501–504. ISBN: 978-1-4244-7279-6.
- [174] H.M. Nam, J.Y. Jeong, K.Y. Byun, J.O. Kim and S.J. Ko. "A Complexity Scalable H.264 Decoder with Downsizing Capability". In: *IEEE Trans. Consum. Electron.* 56.2 (May 2010), 1025–1033.
- [175] J.R. Leonardi. "Multiple time-scales of language dynamics: An example from psycholinguistics". In: *Taylor & Francis, Ecological Psychology* 22.4 (Nov. 2010), pp. 269–285.
- [176] C. Po-Wei and O. Hsin-Ying. *Patent Application – Image processing method and device for performing mosquito noise reduction.* US20080085059A1, Oct. 2006.
- [177] Y. Nie, H-S Kong, A. Vetro and K.E. Barner. "Fast Adaptive Fuzzy Post-Filtering for Coding Artifacts Removal in Interlaced Video". In: Mar. 2005, pp. 993–996.
- [178] J. Singh, D. Singh and M. Uddin. "Detection methods for blocking artefacts in transform coded images". In: 2014, pp. 435–444.
- [179] U.S. Mohammed. "A pixel-domain post-processing technique to reduce the blocking artifacts in transform-coded images". In: *IEEE Int. Symposium Signal Process. Inf. Technol. (ISSPIT)*. 2010, pp. 266–270.
- [180] H. Yao-Min, J. Leou, and M. Cheng. "A Post Deblocking Filter for H.264 Video". In: *Proc. 16th Int. Conf. Computer Commun. Networks (ICCCN)*. 2007, pp. 1137–1142.
- [181] K. Minoo and T.Q. Nguyen. "A perceptual metric for blind measurement of blocking artifacts with applications in transform-block-based image and video coding". In: *Proc. 15th IEEE Int. Conf. Image Process. (ICIP)*. 2008, pp. 3152–3155.

- [182] C. Mantel, P. Ladret and T.Kunlin. "A temporal mosquito noise corrector". In: *Int. Workshop on Quality of Multimedia Experience (QoMEx)*. 2009, pp. 244–249.
- [183] Wenwen Wang Zhenxing Qian and Tong Qiao. "An Edge Detection Method in DCT Domain". In: *Procedia Engineering* 29.0 (2012). Int. Workshop Inf. Electron. Eng., pp. 344 –348. ISSN: 1877-7058.

Acronyms

AVC	Advanced Video Coding
AGWN	Additive White Gaussian Noise
BD	Blu-ray Disc
CPI	Characteristic Point Information
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
DAB	Digital Audio Broadcasting
DCT	Discrete Cosine Transform
DSP	Digital Signal Processor
DTS	Decoding Time Stamp
DTV	Digital Television
DVB	Digital Video Broadcasting
DVB-C	Digital Video Broadcasting Cable
DVB-H	Digital Video Broadcasting Handheld
DVB-S	Digital Video Broadcasting Satellite
DVB-T	Digital Video Broadcasting Terrestrial
DVD	Digital Versatile Disc
ES	Elementary Stream
ETSI	European Telecommunications Standards Institute

ACRONYMS

FEC Forward Error Correction

GA Grand Alliance

GOP Group Of Pictures

HD High Definition

HDD Hard Disk Drive

HDTV High Definition Television

IEC International Electrotechnical Commission

ISO International Organization for Standardization

ITU-T International Telecommunication Union

JPEG Joint Picture Expert Group

JTC1 Joint Technical Committee for Information Technology

JVT Joint Video Team

LCD Liquid Crystal Display

LPF Low Pass Filter

LUT Look Up Table

MC Motion Compensation

MCTNR Motion Compensated Temporal Noise Reduction

MPE Multi-Protocol Encapsulation

MPEG Motion Picture Expert Group

NAL Network Adaptation Layer

OLED Organic Light Emitting Diodes

OSD On Screen Display

OSI Open System Interconnect

PAT Program Association Table

PH Packet Header

PHY Physical Layer

PMT Program Map Table
PCR Program Clock Reference
PCM Pulse Code Modulation
PDP Plasma Display Panel
PES Packetized Elementary Stream
PS Program Stream
PSI Program Specific Information
PTS Presentation Time stamp
PVR Personal Video Recording
RS Reed-Solomon
QoE Quality of Experience
QoS Quality of Service
SD Standard Definition
SI Service Information
SSD Solid State Disc
SVCD Super Video Compact Disc
TNR Temporal Noise Reduction
TS Transport Stream
UHDTV Ultra High Definition Television
VCD Video Compact Disc
VCEG Video Coding Experts Group
VLC Video Coding Layer

“Je pense donc je suis”

René Descartes

Acknowledgements

After a long journey, which was initiated more than a decade ago, this thesis has finally come to an end. Although this work carries my name, it would not have been possible without the aid of many people.

I hereby thank the members of the promotion committee for their time and effort in reading this thesis and their valuable feedback.

First of all, I would like to thank Prof.dr.ir. P.H.N. de With. Dear Peter, the past two decades, you influenced twice my career in a substantial manner. Our first encounter was in 1992, resulting in my involvement in digital recording at Philips Research. Our second encounter, was in 2002, where you challenged me to conduct an industrial PhD. Although the first conference contribution was quickly obtained, the remaining contributions required substantial more time, due to organizational changes. I have a deep appreciation for your devotion, which you showed during the many hours of reviewing, either in the office or at home, during working days or in the weekends and even during holidays. Furthermore, I would like to thank Dr. E. Persoon, for granting this journey in 2002. Moreover, I would like to thank S. Borgers, J. Misker and E. Lamberts for enabling all the involved conference travels.

The work presented in this thesis has been established during various projects, conducted for different organizations involving Philips Semiconductors, NXP Semiconductors and Trident Microsystems. I express my gratitude to Paul van Niekerk and Ewout Brandsma, which enabled the work on video navigation. Furthermore, I would like to thank Armand Stuivenwold, Arie Koppelaar, Edwin Dilling, Frans van de Pavoort and Marc Klaassen for a great TV-on-Mobile project, resulting in an improved DVB-H link layer. Finally, words of appreciation go to Erwin Bellers, Jeroen Kettenis, Paul Hofman, Bahman Zafarifar and Yuanjia Du, for the work on artifact-location detection.

Furthermore, I want to thank Thijs Withaar, Marcel Steenhuizen, Klaas de Waal, Johan van der Graaf and Dr. Stan Baggen for joining the weekly sport activity.

Besides the people with whom I used to work in industry, I also acknowledge and say thanks to the people of SPS-VCA, Egor Bondarev, Sveta Zinger,

ACKNOWLEDGEMENTS

Gijs Dubbelman, Lykele Hazelhoff, Ivo Creusen, Solmaz Javanbakhti, Kostas Triantafyllidis, Fons van der Sommen and of course Anja de Valk-Roulaux.

I also highly appreciate the cooperation of Jan Reinhard en Chrétien Bergmans and the Mechatronic team of Avans University of Applied Sciences Breda, for enabling the final part of this thesis.

Finally, I give special words of gratitude to my familie Maïke, Frederieke, Christoph and my parents for all the patience, especially the last 5 months.

Curriculum vitae



Onno Eerenberg was born in Zwolle, the Netherlands, in 1966. He graduated in Electrical Engineering at the Polytechnical College of Amsterdam, (B.Sc. degree) in 1992 and obtained a MSc. degree in 1998 in engineering product design from the University of Wolverhampton, (UK). He started his career as a research assistant at Philips Research Laboratories working on magnetic recording systems. In this period, he was involved in the hardware realization of digital tape-based recording systems and the development of trick-play for Digital Video Home System (DVHS). In 1998 he joined Philips Communication and Security Systems, where he worked on digitizing the CCTV applications on the basis of MPEG-2 compression and ATM technology. In 2000 he joined Philips Semiconductors where he developed advanced video navigation methods for hard disk recording. In 2003 he returned to Philips Research where he was involved in the development of an efficient and robust DVB-H link layer. In 2007 he switched to NXP Semiconductors Research where he developed video algorithms for digital television and set-top box products. This work was continued in 2010, when Trident Microsystems acquired the BU-Home activities from NXP Semiconductors. Mr. Eerenberg holds 20 patent families (clustered patents) and published 13 papers in the field of digital recording, mobile television reception and television. Furthermore, he is co-author of 2 book chapters, one on DVB-H link layer published by Springer Verlag 2008 and a chapter on Digital Video, published by Intech in 2010. Since 2008 he has served as a Technical Program Committee (TPC) member for the International Conference on Consumer Electronics (ICCE). He is currently a lecturer Electrical and Mechatronic Engineering at Avans University of Applied Sciences Breda, the Netherlands.