Mid-Level Content-Based Video Coding using Texture Analysis and Synthesis

von Diplom-Ingenieur
Patrick Ndjiki-Nya

von der Fakultät IV – Elektrotechnik und Informatik
der Technischen Universität Berlin
zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften
–Dr.-Ing.–

genehmigte Dissertation

Promotionsausschuss:

Vorsitzender: Prof. Dr. Felix Wichmann
Gutachter: Prof. Dr.-Ing. Thomas Sikora
Gutachter: Prof. Dr.-Ing. Jens-Rainer Ohm

Tag der wissenschaftlichen Aussprache: 22.5.2008

Berlin 2008

D83

To Eugène and Hannah

Acknowledgments

I would like to thank Professor Thomas Sikora for his spontaneous willingness to supervise this thesis. My gratitude goes to Professor Peter Noll for advising me to apply at the Heinrich-Hertz-Institut and unconditionally supporting me in numerous administrative procedures, which reminded me of Kafka's novel "The Trial", but this is a different story. Professor Ohm gave me the opportunity to conduct research on thrilling digital image processing fields at the Heinrich-Hertz-Institut and always showed a lot of concern for the well-being of his staff. Accept my gratitude. Many thanks to Dr. Wiegand for believing in my work and giving me helpful advices.

I thank Eugène and Hannah Ndjiki-Nya, my parents, for being there for me, supporting me financially and emotionally, and devoting so much time and energy to my education. The memory of the warm home they offered me and the wealth of advices they gave me throughout helped me to face the challenges and adversities of the past years. Thanks to my wife, Anke, for supporting me, taking the daily constraints from me to keep me focused, baring my moods, giving me the energy to go on, and, last but not least, offering me three healthy children, Frédéric, Félix, and Fanya. Special thanks to my mother and my father-in-law, Barbara and Hanns-Peter Beier, for their invaluable support.

My colleagues Tobias Hinz, Aljoscha Smolić, Béla Makai, Heiko Schwarz, and Peter Eisert deserve particular thanks for their active support in the successful realization of this thesis. Special thanks to Dr. Ralf Schäfer, my head of department, for his admirable understanding of leadership.

Last but not least, I would like to thank my students Oleg Novychny, Christoph Stüber, and Mikel Barrado for their high identification with their respective subjects. This thesis has significantly benefited from your hard work and your critical minds.

Table of Contents

Glossary

Common Notation

(x,y)	Continuous spatial coordinates (width,height)
(ξ,η)	Discrete spatial coordinates (width,height)
(M,N)	Picture resolution (width,height)
t	Time
$m(\xi,\eta,t)$	Gray scale or binary segmentation mask
$I(\xi,\eta,t)$	Luminance channel of arbitrary video sequence
(v_x, v_y)	Motion vector components
τ	Iteration step
N	Arbitrary neighborhood of a pixel or a set of pixels
$N_8(\xi,\eta,\ell)$	8-neighbor set of the pixel at location (ξ,η,ℓ)
σ	Standard deviation
μ	Mean value
a_p	p^{th} parameter of arbitrary motion model
$\mathbf{p} = (a_1,\ldots,a_8)$	(Perspective) motion parameter vector
$\mathbf{k}$	Measurement vector
$\mathbf{D}$	Data-dependent matrix for model fitting
$\boldsymbol{\eta}$	Measurement and modeling error vector
$\mathbf{e}$	Estimation error vector
R	Arbitrary texture region
S	Set of regions or corresponding contours
$\lvert\ \rvert$	Size of a set
$\lVert\ \rVert$	Arbitrary Similarity measure
E	Cost function

Spatial Texture Analysis

$g(\xi,\eta,\ell)$	Picture matrix at quad-tree level ℓ
$\mathbf{h}^{\tau}(u)$	Gray level histogram
W_u^{τ}	Sliding window at bin location u
m	Sliding window size parameter
B_ℓ^r	"Reduced" boundary region at quad-tree level ℓ
B_ℓ^N	8-neighborhood of B_ℓ^r
C	Set of cluster centroids sorted in ascending order
$\mathbf{\Sigma}$	Covariance matrix
$\mathbf{f}_p$	p^{th} eigenvector of $\mathbf{\Sigma}$
λ_p	p^{th} eigenvalue of $\mathbf{\Sigma}$

Spatio-Temporal Texture Analysis

E_D	Data constraint cost function in optical flow estimation
E_S	Coherence constraint cost in optical flow estimation
λ_D	Regularization parameter in optical flow estimation
O_p	p^{th} outlier set (motion splitter)
T_{ms}	Outlier threshold (motion splitter)
μ_e^p	Mean modeling error (M-estimation) of p^{th} segment
$\mathbf{f}_{scc}^p$	Scalable color feature vector of p^{th} segment
T_{scc}	Similarity threshold (temporal mapping)
$\mu_e^{no\_tm}$	Mean error to picture pair before temporal mapping
μ_e^{tm}	Mean error to picture pair after temporal mapping
R_{no_tm}, R_{tm}	Regions before and after temporal mapping (tm)
G_b	Boundary ground truth set
S_b	Boundaries in segmentation mask of system under test
e_b^m, e_b^f	Missing (m) and false (f) boundary error rate

S_b^{tp}, S_b^{fp}	Sets of true and false positive contour pixels
r_b^{tp}	True positive boundary rate
w_b^m, w_b^f	Weight to the missing (m) and false (f) boundary rate
$d_H(S_1 \rightarrow S_2)$	Hamming distance
G_r	Segment ground truth set
S_r	Segments in segmentation mask of system under test
e_r^m, e_r^f	Missing (m) and false (f) segment error rate
r_r^{tp}	True positive segment rate

Texture Synthesis

T	Infinite texture sample
L	Image or video lattice to be synthesized
$\mathbf{t}_i$	Texture pattern extracted from the texture sample T
M	Set of sub-textures $\mathbf{t}_i'$ with a perceptually similar neighborhood to the unknown sample $\mathbf{t}_i$
$G(Nd,E)$	Flow network
Nd	Set of nodes in flow network $G(Nd,E)$
E	Set of edges in flow network $G(Nd,E)$
$K(E)$	Set of weights corresponding to E
T_{src}	Node subset assigned to the source node after min-cut
T_{snk}	Node subset assigned to the sink node after min-cut
E_{cut}	Set of cut edges
$\mathbf{l}_i$	Texture pattern extracted from the synthetic lattice L
$\mathrm{E}_{\mathrm{cut}}(s,p,A,B)$	Cost function of min-cut algorithm
$\varphi_8^x(x,y), \varphi_8^y(x,y)$	Perspective motion model

Video Quality Assessment

$Q_s(t)$	Quality measure for a single picture at time instance t
Q_s	Spatial quality measure for a sequence of pictures
$Q_t(\Delta t)$	Slice-independent temporal quality measure for two consecutive pictures of a video sequence
$Q_t^{\xi}(\Delta t), Q_t^{\eta}(\Delta t)$	Slice orientation dependent temporal quality measures for two consecutive pictures of a video sequence
Q_t	Slice-independent temporal quality measure for a video sequence
$E_o(t), E_d(t)$	Mean occurrence of given gradient directions in the original (o) and distorted (d) signals
$f_\beta(\xi,\eta)$	Linear gradient filter for spatial edge detection
$f_\beta(\xi,t), f_\beta(\eta,t)$	Linear gradient filters for temporal edge detection
$\mathbf{C}(\xi,\eta,t)$	Object contour matrix
$s_\xi(\xi,\eta,t)$	ξ-t slice
$s_\eta(\xi,\eta,t)$	η-t slice
$s'_\xi(\xi,\eta,t,\beta)$	Filtered (orientation $\beta°$) ξ-t slice
$s'_\eta(\xi,\eta,t,\beta)$	Filtered (orientation $\beta°$) η-t slice
$GMA_\xi(\Delta t,\beta)$	Directional (orient. $\beta°$) global motion activity (ξ-t slice)
$GMA_\eta(\Delta t,\beta)$	Directional global (orient. $\beta°$) motion activity (η-t slice)
$GMA_\xi(\Delta t)$	Isotropic global motion information (ξ-t slice)
$GMA_\eta(\Delta t)$	Isotropic global motion information (η-t slice)
$LMA_\xi(\xi,\eta,\Delta t)$	Isotropic local motion information (ξ-t slice)
$LMA_\eta(\xi,\eta,\Delta t)$	Isotropic local motion information (η-t slice)
r_p	Pearson's correlation coefficient
r_o	Outlier ratio

Abbreviations and Acronyms

ADSL	Asymmetrical Digital Subscriber Lines
AUC	Area Under Curve
CABAC	Context Adaptive Binary Arithmetic Coding
CAVLC	Context Adaptive Variable Length Coding
CBIR	Content-Based Image Retrieval
CBVR	Content-Based Video Retrieval
CD	Change Detection
CDM	Change Detection Mask
CIF	Common Intermediate Format
DMOS	Differential Mean Opinion Score
DMOSp	Predicted Differential Mean Opinion Score
DSCQS	Double Stimulus Continuous Quality Scale
DSIS	Double Stimulus Impairment Scale
DSL	Digital Subscriber Lines
GMC	Global Motion Compensation
GoB	Group of Bursts
GoP	Group of Pictures
IEC	International Electrotechnical Commission
ISO	International Organisation for Standardisation
ITU	International Telecommunication Union
JVT	Joint Video Team
LMC	Local Motion Compensation
MPEG	Moving Picture Experts Group
MRF	Markov Random Field
MSE	Mean Squared Error
NAL	Network Abstraction Layer
ORDF	Operational Rate-Distortion Function
PCA	Principal Component Analysis
PPS	Picture Parameter Set
PSNR	Peak Signal-to-Noise Ratio
QCIF	Quarter Common Intermediate Format
QP	Quantization Parameter

ROC	Receiver Operating Characteristic
SFM	Structure From Motion
SH	Slice Header
SPS	Sequence Parameter Set
SSCQE	Single Stimulus Continuous Quality Evaluation
TSCES	Triple Stimulus Continuous Evaluation Scale
VCEG	Video Coding Experts Group
VCL	Video Coding Layer
VDSL	Very high speed Digital Subscriber Lines
VQA	Video Quality Assessment
VQEG	Video Quality Experts Group
VQM	Video Quality Measure

1 Introduction

Content-based representation and coding of visual information has been a very active research area in the past decade. This is mainly due to the MPEG-4 [1] and MPEG-7 [2] standardization activities. MPEG, which stands for Moving Pictures Experts Group, is a working group of the ISO/IEC (International Organisation for Standardisation/ International Electrotechnical Commission) that develops standards for entertainment, consumer electronics and similar applications.

MPEG-7 visual addresses the content-based description of visual information to allow efficient access to multimedia material for searching, processing and filtering purposes. MPEG-7 provides a set of standardized tools to describe multimedia content, i.e. the representation of the content description is normatively specified, while the generation and utilization of the same is not. For that, MPEG-7 can be used for a large range of applications.

MPEG-4 follows an object-based coding approach, where a so-called video object can have either rectangular or arbitrary shape. The standard thus encompasses a complementary model to pixel-based or block-based visual information descriptions, which enables interaction and hyperlinking capabilities as well as composition of natural and synthetic video objects into a hybrid scene context. This is achieved by representing objects in a given scene independently and supporting individual access for their manipulation and re-use. Hence, tools for applications such as interactive video services can be found in the MPEG-4 framework. The object-based coding features of MPEG-4 have however experienced very limited success as far as applications relating to natural video sequences are concerned. This is because the automatic identification of video objects in generic video scenes is a non-trivial task. Hence, to date, known automatic scene analysis and segmentation algorithms are not capable to guarantee the accuracy required to run an MPEG-4 codec. In the MPEG-4 framework, the video objects to be processed can be of rectangular shape as mentioned above. Recently, a hybrid, block-based version of MPEG-4 known as H.264/MPEG4-AVC [3] was released. It is a new part of the standard and is

neither backward nor forward compatible with earlier standards in order to achieve highest possible compression performance.

The purpose of the content-based video coding approach presented in this thesis is to achieve bit rate reduction of compressed video material, while preserving high visual quality of decoded sequences. It is shown that content-based features can be integrated into a block-based, hybrid video codec as H.264/MPEG4-AVC with significant coding efficiency gains and less constraining requirements w.r.t. video object boundaries compared to MPEG-4 with arbitrary shape coding.

1.1 Objectives

The present work aims to show that the Mean Squared Error (MSE) criterion, typically used in hybrid video codecs as H.264/MPEG4-AVC [3], is not an adequate coding distortion measure for high frequency regions displayed with limited spatial resolution. The relevance of these textures is given by the fact that they typically yield high coding costs. It shall be shown that global similarity measures are better suited for assessing the distortion of such textures, as no MSE-accurate reconstruction of the latter is required. Coding gains as well as limitations of the proposed approach are documented in this work.

The above-mentioned concept is adapted to a new, generic, and fully automated content-based video coding scheme. The latter provides a texture analyzer at the encoder and a texture synthesizer at the decoder. The texture analyzer thereby identifies the target textures, i.e. highly texturized regions that are displayed with restricted spatial accuracy. For these textures, it is assumed that the viewer perceives the semantic meaning of the displayed texture rather than the specific details therein. The texture synthesizer regenerates an approximate version of the original texture based on corresponding meta data transmitted by the encoder. The texture itself is not coded in the proposed framework, which is designed to be closed-loop, i.e. a mechanism to identify and where necessary alleviate artifacts due to erroneous analysis or synthesis is required, in order to generate controlled (subjective) video quality at the decoder end.

A set of effective tools are tailored for the proposed framework in order to achieve the expected coding gains w.r.t. H.264/MPEG4-AVC without the proposed

approach. As mentioned before, video quality assessment tools for detection of possible spatial or temporal impairments in the reconstructed video are required to meet with the above-mentioned quality constraints. The texture synthesizer must be adapted to constrained synthesis of spatio-temporal textures given a moving camera, where the spatial and temporal constraints are given by adjacent natural textures. The video coding framework presented in this thesis is integrated into an H.264/MPEG4-AVC video codec and should yield at least the same performance as the codec without the proposed approach, w.r.t. bit rate and video distortion. A rate-distortion decision module that optimizes the side information that is transmitted to the decoder ensures this. That is, textures for which no rate-distortion gains can be achieved, compared to the genuine H.264/MPEG4-AVC codec, are coded by the latter. That is, the genuine codec acts as fallback coding solution.

The approach presented in this thesis is designed to be robust, where robustness is essentially twofold in this framework:

- The first aspect addresses cases where a target texture turns into a non-target texture in the course of a video sequence. This specific situation arises, for instance, when a zoom occurs and a texture is shown with much higher spatial accuracy than before. The system must then seamlessly switch between a synthesized and an MSE-accurate texture representation.
- The second aspect concerns eventual infinite synthesis artifact propagation.

In the following sections, the main contributions of the present thesis as well as its organization will be presented.

1.2 Main Contributions

The main contributions of this thesis can be summarized as follows:

1) **The framework**: A modular and thus flexible content-based video coding framework that can be easily integrated into any standards-compliant video codec is presented. Within this framework, simple replacement of any component of the scheme can be operated.

2) **Bit rate savings**: The required bit rate for transmitting highly texturized regions can be significantly reduced, if the number of bits needed for their approximate description using a global distortion measure is smaller than the number of bits for the description using MSE. In this thesis, it is shown that bit rate savings up to 41%, at the same visual quality, can be achieved compared to H.264/MPEG4-AVC [3] without the presented algorithm.

3) **Spatio-temporal texture analysis**: A new spatio-temporal texture analysis algorithm is presented. It is based on robust statistics, namely an M-estimator [4], and incorporates an MPEG-7 [2] descriptor for consistent temporal labeling of identified textures.

4) **Precision requirements w.r.t. texture analysis**: The presented framework enables reduced precision requirements towards the texture analysis component, as the pixel-accurate analysis result of the latter is mapped onto the macroblock grid prescribed by hybrid video codecs like H.264/MPEG4-AVC. Hence, artifact probability is considerably reduced compared to object-based video coding approaches where pixel-precise shape information has to be transmitted to the decoder. The scheme introduced in this thesis features nearly a full macroblock tolerance w.r.t. the homogeneous region boundaries. Motion boundaries are left to the fallback codec (e.g. H.264/MPEG4-AVC) to encode.

5) **Texture synthesis**: A generic texture synthesis approach is presented. It is a non-parametric, patch-based synthesis algorithm and for that applicable to a large class of spatio-temporal textures with and without local motion activity.

While stuffing a texture in a video sequence, spurious edges are avoided by using graph cuts to generate irregular contours at transitions between natural and synthetic textures and place them (the contours) in high frequency regions, if any, where they are less visible.

A new synthesis algorithm optimized for rigid textures is further introduced in this thesis. It features better compression properties for the given texture class as it requires significantly less key pictures than the synthesizer described above. The rigid texture synthesizer is based on global motion compensation of texture segments that are automatically identified by the spatio-temporal texture analysis module. The former comprises a post-processing operation of the segmentation masks that determines the maximum synthesizable texture area without significant degradation of the quality of the synthesis result.

The texture synthesis algorithms are finally integrated into the video coding framework of the present thesis. A compact side information is thus defined for each of the synthesizers for transmission from the encoder to the decoder.

6) **Video quality assessment**: New Video Quality Assessment tools that ensure artifact-free video at the decoder end are presented in this thesis. Spatio-temporal video impairments related to video synthesis are explicitly detected and tackled. The proposed VQ measures feature high correlation with the subjectively perceived video quality by human subjects. The complexity of the former is significantly lower than for VQMs entailing comparably high correlations.

1.3 Achievements

The video codec proposed in the present thesis has led to six patent applications to date of which one has already been granted. The titles of the patents applied for are:

1) Artifact detection in pictures and video sequences (08/2007, German patent, pending),
2) Concept for determining a video quality measure (03/2007, international/PCT patent, pending),
3) Concept for synthesizing texture in a video sequence (03/2007, international/PCT patent, pending),
4) Approach for intelligent scaling of single pictures and video sequences (12/2006, German and US patents, pending),
5) Video coding with texture analysis and synthesis I (02/2003, European patent, granted),
6) Video coding with texture analysis and synthesis II (02/2003, European patent 1599835, granted).

1.4 Organization of the Thesis

The remainder of the thesis is organized as follows. In Chapter 2, the content-based video coding framework is presented in detail. A discussion of relevant approaches is proposed. For that, content-based coding schemes are subdivided into low-, mid- as well as high-level schemes. The characteristic features of each of the above-mentioned categories are presented. Additional approaches showing a sufficient relationship to the framework presented in this thesis are also reviewed.

In Chapter 3, the principle of the proposed framework is presented. The important notion of texture is defined. Assumptions made in the given framework are motivated. The overall framework is described into detail as well. The modules required in the proposed framework are introduced, their necessity illustrated, and their interactions explained.

In Chapters 4 to 6, the texture analyzer, the texture synthesizers, and the video quality assessors are presented into detail. For each of these modules, the state-

of-the-art is described and the proposed methods are discussed in the light of known efficient techniques. Comparative performance evaluations are conducted to document the improvements achieved by the new methods compared to state-of-the-art approaches.

In Chapter 7, an in-depth description of the interactions between the modules in the overall framework is given, while the rate-distortion decision module is presented in Chapter 8.

In Chapter 9, the integration of the proposed approach into an H.264/MPEG4-AVC video codec is discussed. The latter is the latest standard developed by the Joint Video Team that was founded by MPEG and VCEG to achieve a significant increase of compression efficiency of existing video codecs. The H.264/MPEG4-AVC standard is also known as ISO/IEC 14496-10 ("Coding of Audiovisual Objects – Part 10: Advanced Video Coding") or ITU-T Rec. H.264 [3]. In this chapter, it is shown that the insertion of the meta data required by the new approach into the codec's bitstream can be done without significant modification of the corresponding syntax and semantic.

In Chapter 10, the experimental results of the overall framework are presented. Video sequences that contain textures (rigid as well as non-rigid) useful to demonstrate that an approximate representation of some textures can be done without subjectively noticeable loss of quality are examined. High-level features of the video codec are thereby enabled and the coding gains achieved by the proposed approach are documented.

2 State-of-the-Art Content-Based Video Coding

Content-based video coding aims to achieve bit rate reduction of compressed video sequences, while preserving high visual quality of decoded data [5]. As opposed to statistical video coding approaches, where statistical features of the video signal are exploited to detect redundancies and thus achieve bit rate reduction, content-based video coding approaches decompose the sequence into spatially, temporally or spatio-temporally coherent regions. The coherence of a region is thereby measured based on motion, color and/or texture features. These object attributes are typically described via compact representations given the video coding framework.

Content-based video coding schemes can be clustered into low-, mid- and high-level techniques based on the semantic significance or insignificance of the objects they are tuned to identify. Systems with the capability of automatically capturing semantically meaningful objects can be seen as high-level approaches. They represent the most challenging content-based video coding class and will probably be beyond reach for several years to come. Mid-level techniques reduce coding costs by processing different regions with similar motion, texture or color characteristics together. That is, for such algorithms, the semantic content of the identified objects is irrelevant. However, these objects must be described consistently in space and time. In this work, low-level approaches will be defined as coding techniques that exclusively rely on spatial or temporal features for the detection of homogeneous regions and do not incorporate any inference mechanism concerning tracking and/or spatial consistency of identified regions.

Most mid- and high-level content-based video coding approaches feature two main steps, an analysis and a synthesis phase. The analysis phase yields an a posteriori decomposition of the video sequence, in contrast to a priori defined regions such as block partitions. The motion model utilized in this phase is crucial in the sense that it affects the complexity of object motion and deformation that can be described. Either 2D or 3D motion models can be used [6],[4]. Given an adequate model, the number of objects can be limited and their temporal evolution tracked. Important phenomena like covering,

uncovering, object appearance and disappearance can thus be addressed purposefully, which enables an enhanced identification and exploitation of scene redundancies with regard to bit rate reduction. The synthesis phase consists in rate-distortion constrained merging of the identified spatio-temporal objects. That is, the challenge consists in finding a partition that minimizes the global distortion given a bit rate budget [7],[8],[9],[10]. Approaches complying with the above-mentioned strategy are also called second generation video coding techniques.

2.1 Mid-Level Schemes

Research on content-based description and coding of video information has been catalyzed by the multimedia applications tackled by the MPEG-4 standard, i.e. interactive video services, interactive graphics etc.. MPEG-4 is a standard developed by the Moving Picture Experts Group (MPEG), a working group of ISO/IEC [1], that standardizes the storage and delivery of multimedia content. In the MPEG-4 framework, an image is assumed to be composed of a finite set of arbitrarily shaped objects. Intrinsic features like shape, texture and motion can describe each of these objects. Separate decoding and reconstruction of objects as well as manipulation of the original video scene by simple operations on the bitstream are enabled. Thus content-based interactivity and compression can be achieved in this framework [1]. A further feature of MPEG-4 video coding can be seen in the content-based scalability, i.e. the ability to distribute the available bit rate between the different objects of the scene [1],[11].

Many mid-level video coding schemes can be found in the literature [12],[13],[14],[15],[16],[17],[18],[19],[9],[20],[21],[22]. They can be categorized by the predominant feature(s) they use for object homogeneity inference. Some approaches [14],[17],[9],[22],[23] utilize spatial features for consistency analysis, while other [13],[12],[15],[16] use motion descriptors as homogeneity criterion. A third set of algorithms [18],[19],[20],[21],[24],[25] fuses both spatial and motion features for object detection. Some of the approaches closely related to this work will be reviewed in detail in the following.

One of the early descriptions of a mid-level video coding approach similar to the one presented in this thesis was published by Adelson [13]. It is assumed that

any video sequence can be represented as a set of overlapping layers, where a layer is the description of a coherent motion region. Ordering the layers in depth and applying the rules of compositing ideally yields the original video sequence. Each layer is extracted through segmentation and described by a set of maps. Wang and Adelson [16] propose three essential maps for compositing the extracted layers: The intensity, the alpha and the velocity map. The intensity map, also called texture map, represents a summarization of the texture appearance over time. The alpha map defines the transparency or opacity at each point. Finally, the velocity map defines how the intensity map should be warped over time to achieve the reconstructed video sequence. The most remarkable feature of Wang and Adelson's approach [16] is the extraction of depth and occlusion information from the masks obtained through robust motion segmentation of the video sequence. Segmented spatio-temporal volumes are warped towards a reference picture yielding stationary regions in the motion compensated sequence given an accurate motion parameter estimation. A single intensity map is subsequently derived from the segmentation masks and the original sequence for each coherent motion region by applying a temporal median filter operator on the motion compensated video. Thus the whole video sequence can be collapsed to a typically small set of intensity maps that can be used, along with the other maps, to regenerate the entire video sequence. A so-called delta map can be used to temporally update the intensity map in order to account for illumination variations for instance. The delta map serves as an error signal [16]. After motion compensated median filtering, occlusion relationships are determined. For that, given neighboring layers, the layer that is derived from more pixels occludes (alpha map) the layer that is derived from fewer pixels. It is assumed here that the reliability of a layer is proportional to the number of pixels it has been derived from (cp. intensity map). The limiting factor of this approach is obviously the analysis of the video into layers. Although it allows for overlapping regions (layered representation), the algorithm presented in [16] inherently requires a very precise segmentation of the video sequence to minimize subjectively annoying artifacts. Furthermore, Wang and Adelson's approach [16] is open-loop, i.e. there is no mechanism to identify and where necessary alleviate artifacts due to erroneous segmentation, which yields unregulated (subjective) video quality at the decoder output.

Yoon and Adelson [21] proposed a texture coding framework both for still and moving images that can be seen as an extension of the work in [16] from the viewpoint of video coding. A video sequence is first decomposed into layers [16]. Each layer is segmented into texturally uniform regions and each region is described by a set of texture parameters derived from Heeger and Bergen's framework [26], i.e. textures are modeled using steerable pyramid subband histograms. The model parameters and shape information are transmitted and used to synthesize and fill in the uniform regions (within a layer) at the decoder side. Yoon and Adelson's approach [21] being a submodule of the approach in [16], the limitations addressed above still hold.

Dumitraş and Haskell proposed a content-based video coding method by texture replacement [22]. Replaceable textures are identified and removed from the corresponding regions of the original pictures. The resulting video sequence is encoded and the extracted parameters of the removed textures transmitted to the decoder. Note that region shape parameters are not coded. At the decoder, the missing textures are synthesized based on the related parameters and inserted into the decoded video. A removable texture is defined as covering at least 40% of the picture area over at least 50 pictures and belonging to the class of "typical movie textures". The texture removal stage consists in region segmentation and texture analysis for parameter extraction. Absent or very slow global motion of removable textures as well as few scene objects with moderate motion are assumed, which are very strong constraints that confine the practical usability of this approach to restricted applications. The approach presented in [22] is pragmatic in the sense that only replaceable textures are synthesized, while the others are left to any standards-compliant video codec to encode. The latter serves as a fallback solution in case texture replacement is not feasible. This represents a significant improvement of [16] from the viewpoint of practicability. In fact, in [16], the attempt is made to implement a self-contained video codec solution based on the analysis-synthesis video coding paradigm. Unfortunately, the limitations due to partially unreliable texture analysis make the emergence of a generic video codec of this type unlikely, at least in the next few years. The approach by Dumitraş and Haskell [22] is open-loop and thus yielding the related drawback described above. Notice that a similar algorithm was recently proposed by Flores and Polon [23].

Sarnoff Corp., Alcatel, SBC Labs and Thomson are currently developing video-over-DSL technology, also called Internet Protocol Television (IPTV), based on a content-based video coding approach. Their objective is to enable video services over the US telephone network, by exploiting the existing Asymmetrical Digital Subscriber Lines (ADSL). It is intended to accommodate two video streams on a single typical DSL link for real-time distribution, decoding and viewing. Hence compression of video sequences to 500kb/s, assuming an ADSL downstream data capacity of 1.5 Mb/s, is required That is 33% less than the H.264/MPEG4-AVC [3] video codec at Digital Television (DTV) quality, where H.264/MPEG4-AVC is a standard jointly developed by MPEG and the International Telecommunication Union (ITU). Movie coding at these bit rates can in principle be achieved with standards compliant video codecs (e.g. H.264/MPEG4-AVC), at the cost, however, of noticeably deteriorated video quality. The consortium's approach basically comprises two key components, namely a Tweening-Based video Compression (TBC) module and an intelligent video multiplexing scheme. TBC is based on a computer animation technique that consists in replacing intermediate pictures of two given key pictures by applying appropriate interpolation algorithms [27]. The key pictures are coded by any standards-compliant video codec, while the "tweens" or "in-betweens" are derived from them. The intelligent video packet prioritization scheme is projected to minimize visual impairments due to discarded packets, when traffic exceeds transmission limits. Hence, a novel encoding and encapsulation technique is required.

Two mid-level, open-loop approaches, inspired by the present thesis, have been recently proposed. They consist in integrating texture analysis and synthesis into an H.264/MPEG4-AVC codec for effective video representation. The first method by Zhu et al. [25] exploits block-based motion information for region removal. 8x8 blocks are classified into structural and textural categories. Structural blocks are defined as containing edges, which are determined using a simple edge detector. Textural blocks are the remainder blocks, where Zhu et al. [25] distinguish between textural blocks in the vicinity of structural ones and other textural blocks. Structural and neighboring textural blocks are H.264/MPEG4-AVC coded, while blocks to be removed are selected among the remaining textural blocks. Removed regions as well as their motion information are not coded. Zhu et al.

present a spatio-temporal texture synthesis algorithm for region recovery in which spatial and temporal smoothness are simultaneously considered [25]. The second scheme by Bosch et al. [24] is based on a texture synthesis method that is similar to the one proposed in Sec. 5.2. Different segmentation algorithms are examined to evaluate their influence on the overall bit rate savings achievable with their codec compared to genuine H.264/MPEG4-AVC at same visual quality.

Automatic, generic and effective object definition remains the greatest challenge in the field of mid-level video coding. Nevertheless can semi-automatic and thus time consuming segmentation techniques be applied where affordable. Alternatively, video objects can be easily extracted by using videos based on blue-screen composition or synthetic sequences. For most video sequences and applications, however, fully automated analysis tools are required.

2.2 Low-Level Schemes

Global Motion Compensation (GMC) tools are very helpful to increase video coding efficiency given sequences with specific characteristics. Thus MPEG-4 [1], for instance, exploits GMC tools to improve video coding performance in terms of both visual quality and compression ratio.

GMC tools perform best for sequences where camera operations like translation, zooming or rotation predominate. Taking advantage of GMC implies the estimation of motion parameters that describe the 2D motion of two consecutive pictures or Video Object Planes according to MPEG-4 terminology. Within the MPEG-4 framework, the motion parameters are applied on predefined pixel sets or macroblocks. Corresponding motion vectors are also estimated at the encoder for Local Motion Compensation (LMC). A mode decision between GMC and LMC is performed given the respective prediction errors. If the GMC mode is selected, the motion vectors typically transmitted by an hybrid video codec [4],[1],[3] are substituted by the estimated motion parameters and the residual prediction error is transferred to the decoder. Due to the integration of GMC into the mode decision process of the encoder and the transmission of the residual error, GMC can be seen as a generic tool. Eventual bit rate gains achieved through GMC are dependent on the amount of motion vectors that can be substituted throughout the video sequence. MPEG-4 also standardizes static

sprite coding functionalities. A sprite is an image typically containing background information visible at least once in a given video segment. Thus the sprite can be used to reconstruct the background of the whole video piece or for predictive coding of the background. Sprites for background are also referred to as "background mosaics" in the literature. Static sprites are generated offline and transmitted at the beginning of each session, i.e. before coding the video segment itself. The sprite is composed using global motion estimation tools. After the sprite is transmitted, only the motion parameters allowing reconstruction of the current picture's background from the mosaic are transferred. Thus, static sprite coding can help improve visual quality at the decoder. However, static sprites are obviously inefficient for online applications. This drawback is alleviated by an additional functionality provided by MPEG-4, i.e. the piecewise transmission of the sprite. The latter purveys the flexibility to transmit the sprite as required by the timing considerations and bandwidth constraints (e.g. streaming applications) as well as to transmit a lower quality version of the sprite initially and improve it over time with residual images.

Smolić et al.[28] introduced a video coding approach, called Long-Term Global Motion Compensation (LT-GMC), that can be seen as a combination of GMC and sprite prediction. In their framework, picture areas that strictly underlie global motion, basically induced by camera operation, are extracted by applying rate-distortion constrained criteria. For these areas, a single motion parameter set per picture, estimated w.r.t. several previously transmitted frames, is transferred to the decoder. No prediction error signal is needed for the identified regions. A set of super-resolution mosaics[28], comparable, in terms of functionality, to the intensity map in [16], is derived from the original video sequence and used to regenerate the above-mentioned areas given the corresponding motion parameters. The significance of the approach by Smolić et al. [28] can be seen in its ability to alleviate spatial alias like motion blur and lowpass effects due to interpolation filters. Thus extensive improvement of prediction results is achieved compared to standard GMC [1]. Comparably to [22], the solution proposed in [28] builds on a standards-compliant video codec, namely H.264/MPEG4-AVC, which task it is to encode the picture areas that violate the underlying assumptions of the proposed approach, i.e. the picture regions exhibiting local motion. The largest bit rate savings are achieved for high-

resolution video sequences with a dominant global motion (camera pan). That is, in video sequences with picture patches undergoing different motions this approach will fail in the sense that no significant bit rate savings will be achieved.

Steinbach et al. [29] introduced the combination of multiple reference frames and affine motion-compensated prediction . In [29], a segmentation-free solution with more than two reference frames is presented. A suitable reference picture for motion compensation is selected using a Mean-Squared-Error-based (MSE-based) cost function as the distortion criterion. For textures with a high amount of shown detail, however, an MSE-precise reproduction can be considered as dispensable if they are shown with limited spatial accuracy and the original video is not known to the viewer. Hence, for such textures, MSE is not suitable for efficient coding, since irrelevant detail may be reproduced.

2.3 Related Fields

Content-based video coding has some common ground with model-based analysis-synthesis coding as well as content-based image and video retrieval approaches.

The model-based video coding scheme exploits a priori knowledge of the video scene to achieve high compression rates [30],[31],[32],[6]. Three-dimensional models (e.g. polygonal wire-frame structures) of the scene objects are built at the encoder to describe the corresponding shapes and textures. Similar models, transmitted only once at the beginning of the session, are also used at the decoder to substitute the objects. Feature points on the latter are then tracked at the encoder, e.g. eye direction or disposition of the mouth in the video telephony framework, and corresponding minor information is transmitted to animate the decoder model through 3D motion and deformation. Striking visual results can be achieved by this approach at very low bit rates. Nevertheless, severe object class and motion constraints apply. The major limiting factor of the model-based coding approach can be seen in the reliability of the analysis, i.e. the robust detection and tracking of feature points.

Content-Based Image/Video Retrieval (CBIR/CBVR) emerged from the swiftly increased interest in the potential of digital images over the last decade, driven

at least in part by the rapid dissemination of imaging in the World-Wide Web. CBIR/CBVR can be regarded as the process of retrieving desired images or videos from a large collection based on low-level, mid-level or high-level features [33],[34],[35],[36],[37]. The utilized descriptors are typically compact content representations that can be extracted automatically from the data. Hence text-based image retrieval is generally not conceived as a CBIR approach, although the keywords describing an image might delineate its content. CBVR features are a superset of CBIR features, the most obvious CBVR-specific cue being the motion information. The features used for content-based video coding are the same or at least derived from CBVR descriptors. Within the CBIR/CBVR framework, descriptors represent meta data that can not be used to reconstruct the original image or video. They rather enable fast content-based search through a large data base given a query. After the query is completed and the desired images or video sequences are identified, the latter have to be accessed by the user for visualization of the genuine data. As opposed to CBIR/CBVR, the compact descriptors are partly used to model and reconstruct the video sequence in the context of content-based video coding.

3 Overall Framework

3.1 Visual Texture: An Ambiguous Notion

The perception of texture is based on luminance and chrominance fluctuations due to the absorption and reflection properties of corresponding object surfaces [4]. No generic definition of the expression "visual texture" can be, however, found in the literature. Proposed definitions are often directly derived from the concrete application aimed at or the features used for image analysis. For instance, the definition underlying the "texture encoding" concept of MPEG-4 Visual is fairly abstract and refers to the entire pixel-based image information including color [1],[4]. The largest cross-section between the different visual texture definitions is obtained by categorizing textures into homogeneous and inhomogeneous ones.

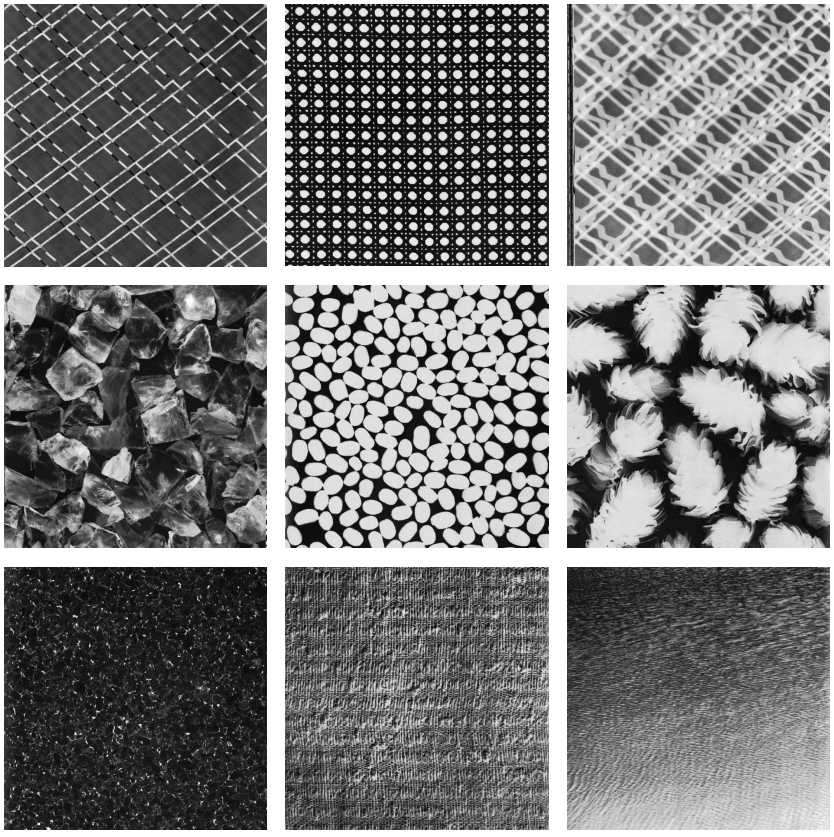


Fig. 1 – Texture examples extracted from the Brodatz [38] texture database. Regular textures (first row), irregular textures (second row), and heterogeneous textures (third row).

Spatially homogeneous visual textures feature some degree of stationarity, i.e. they have a spatially uniform distribution of local color variations. Repeated basic primitives of a visual texture can be referred to as texture elements or texels. The latter are typically subject to randomization of their location, size, and orientation in natural images [39]. The degree of texel randomization yields regular (low randomization) to irregular (strong randomization) textures, where the transition between these two extremes is seamless. Regular textures are shown in the first row of Fig. 1. It can be seen that they have a periodic or quasi-periodic structure. Although the irregular textures, shown in the second row of Fig. 1, feature dominant texture elements, no obvious periodicity can be observed. Some heterogeneous textures are shown in the third row of Fig. 1. As can be seen, such visual textures are characterized by an instationary behavior.

In this thesis, the expression "visual texture" is employed indifferently for homogeneous (e.g. waves, flowerbed) and inhomogeneous (e.g. sand, grass) textures. Textures showing abrupt spatial or temporal dark-bright transitions are referred to as high frequency textures, while those with more gradual dark-bright transitions are called low frequency textures in this work. The concept of "abruptness" and "gradualness" is thereby empirical.

3.2 Assumptions

In this thesis, a generic, closed-loop, mid-level content-based video coding scheme is proposed. It is assumed that many video scenes can be classified into detail-relevant and detail-irrelevant textures. Here, detail-irrelevant textures are defined as highly texturized regions that are displayed with restricted spatial accuracy, while the other textures are referred to as detail-relevant. It is further assumed that for detail-irrelevant textures, the viewer perceives the semantic meaning of the displayed texture rather than the specific details therein.

Many detail-irrelevant texture regions are costly to code, when using the mean squared error criterion as the coding distortion. Thus, in this thesis, it is argued that MSE is not an adequate distortion measure for efficient coding of detail-irrelevant textures and it is claimed that global similarity measures, e.g. MPEG-7 [2] descriptors, are better suited for assessing the distortion of such textures, as no MSE-accurate regeneration of these is requested. Often, the required bit rate

for transmitting detail-irrelevant textures can be significantly reduced, if the number of bits needed for their approximate description using the modified distortion measure is smaller than the number of bits for the description using MSE.

3.3 Overall System

3.3.1 General Description

In this thesis, a content-based video coding method is proposed. The corresponding block diagram is depicted in Fig. 2. As can be seen, the proposed system comprises several sub-modules, i.e. a Texture Analyzer (TA), two Side Information (SI) generators, a Quantizer (Q), two Texture Synthesizers (TS), two Video Quality Assessors (VQA), a State Machine (SM), and a Rate-Distortion Decision (RDD) module. These modules have been developed within the present thesis and some of them constitute efficient tools in their own right. This particularly applies to the spatio-temporal texture analyzer (cp. Chapter 4), the texture synthesizers (cp. Chapter 5), and the video quality assessors (cp. Chapter 6). In the course of the integration of these modules into the video coding system, the above-mentioned tools have been, in some cases, adapted to meet the requirements of the given framework, as documented in the corresponding sections.

Many mid-level video coding schemes can be found in the literature as explained in Sec. 2.1. Most of them are open-loop, i.e. there is no mechanism to identify and where necessary alleviate artifacts due to erroneous analysis or synthesis, which yields unregulated (subjective) video quality at the decoder end. The system proposed in this thesis accounts for this major flaw in that it is a closed-loop method.

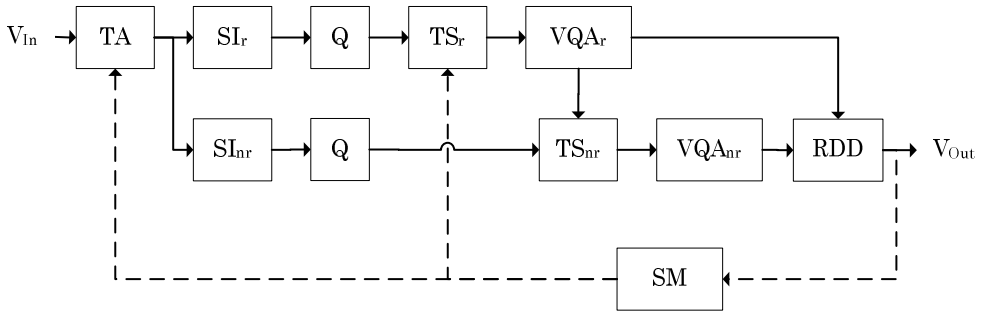


Fig. 2 – Principle of the closed-loop analysis-synthesis video coding approach

In the proposed closed-loop method, the incoming video sequence is evaluated by the texture analyzer that identifies potential detail-irrelevant textures. These textures are discriminated using color and motion features, as well as robust statistics (cp. Chapter 4). Given the detail-irrelevant texture candidates, the side information generators retrieve the required (quantized) meta data. The texture synthesizers to generate synthetic versions of the original detail-irrelevant textures utilize these data. The two synthesizers differ in the type of texture they have been optimized for. TS_r is a rigid texture synthesizer (e.g. flowers, sand, etc.), while TS_{nr} is primarily used for non-rigid texture synthesis (e.g. water, smoke, etc.). As can be seen in Fig. 2, TS_r is executed first, while TS_{nr} is only executed if TS_r fails to synthesize the given texture. Although each of the texture synthesizers requires specific side information, they both have in common that they represent synthesis by example approaches. That is, for each detail-irrelevant texture to be synthesized, they require a representative texture reference for successful synthesis. Hence, the incoming video sequence is divided into Groups of Pictures (GoP). Rigid texture GoPs comprise two reference pictures, the temporally youngest and oldest pictures, and an arbitrary number of partially synthesized pictures in-between (cp. Sec. 5.2) that are synthesized independently of each other. As non-rigid textures are volumetric, i.e. 3D textures, corresponding GoPs comprise two reference picture bursts, the temporally youngest and oldest pictures, and a partially synthesized picture burst in-between (cp. Sec. 5.3), where several pictures are synthesized simultaneously. Hence, a non-rigid GoP encloses several rigid ones. The reference pictures (bursts) hold the texture examples and are thus not synthesized. The synthesized (rigid or non-rigid) GoP is subsequently submitted to the video quality assessment unit for detection of possible spatial or temporal

impairments in the reconstructed video. In the subsequent iterations, a state machine explores the degrees of freedom of the system for generation of relevant side information options. Once all relevant system states have been visited for the given input GoP, a rate-distortion decision is made and the optimized side information is transmitted to the decoder.

The closed-loop character of the proposed approach is given by the fact that a given synthesis path (SI → Q→ TS → VQA) is repeated until the synthesis quality is validated by the VQA or the given side information is rejected because the maximum number of iterations has been reached. However, this is not depicted in Fig. 2 for legibility reasons. The VQA modules modify the masks showing potential detail-irrelevant textures (initially generated by the TA) at each iteration. An in-depth technical description of the interaction between the modules of the proposed content-based video coding system can be found in Chapter 7.

3.3.2 Video Coding Framework

The proposed video coding scheme is integrated into an H.264/MPEG4-AVC video codec. The latter is a hybrid codec, as it exploits both spatial and temporal dependencies of the source signal. Roughly speaking, H.264/MPEG4-AVC defines three picture types of interest for the present thesis, i.e. Intra (I), Predictive (P) and Bi-predictive (B) pictures. These are split into macroblocks. A given intra macroblock is determined using only (encoded, decoded and reconstructed) information from the currently processed macroblock, while predictive macroblocks are determined through motion-compensated prediction using one or more previously decoded reference pictures. The major difference between P and B pictures resides in the way the prediction signal is built from the reference pictures. The reader is referred to Chapter 9 for a more detailed description of the H.264/MPEG4-AVC video codec.

The first non-rigid GoP consists of the first I picture of the sequence and the last picture of the GoP is the first P picture. Between these I and P pictures are B pictures. For example, when 3 B pictures are used, the first GoP has the structure IBBBP$_1$ in temporal order. The second GoP consists of the last picture (the P$_1$ picture) of the first GoP and the next P picture. That is, the second GoP

has the structure P_1BBBP_2. I and P pictures are used as key pictures by TS_r such that corresponding textures are coded using MSE distortion and an H.264/MPEG4-AVC encoder (cp. Sec. 5.2). B pictures (between the key pictures) are candidates for a possible partial texture synthesis and are also otherwise coded using MSE distortion and H.264/MPEG4-AVC. TS_{nr} on the contrary requires several rigid GoPs in order to enable non-rigid texture synthesis as already mentioned above (cp. also Sec. 5.3).

Notice that the side information generation depicted in Fig. 2 is part of the texture analysis and is singled-out in this figure for better legibility of the system. Detail-irrelevant textures for which no rate-distortion gains can be achieved are coded by the reference codec, which acts as fallback coding solution. Furthermore, the GoP structure used in this framework is overlapping, which allows a seamless change from detail-irrelevant to detail-relevant coding, as the key pictures (bursts) are coded based on MSE. In the following, the modules of the new content-based video coding approach are explained in detail.

4 Texture Analyzer

Texture analysis typically requires segmenting an image or video sequence into uniformly textured regions. The segmentation step is both critical and essential, as its accuracy has a significant impact on the quality of the final analysis result. Unfortunately, image (sequence) segmentation is also one of the most difficult tasks in image processing.

For spatial segmentation, the complexity of the texture analysis task mainly arises from two problems [40]. The first difficulty lies in the inherent variability of natural textures, which has implications for any local texture measure used in the context of texture analysis. In fact, region boundaries must be identified without making too strong and thus unrealistic assumptions with regard to the homogeneity of the texture within a given region. The second difficulty is called uncertainty problem and can be described as follows: To achieve a detailed representation of gray value discontinuities, high spatial resolution is required. However, given the latter input, boundary detection methods are likely to generate spurious responses within textured regions. On the other hand, in order to segment textured images into meaningful regions, a sufficiently large area averaging process is needed to reduce the fluctuations in texture properties. Although increasing texture regions' homogeneity, this lowpass operation yields blurred boundaries. Obviously, the uncertainty problem arises from the conflict between the simultaneous measurement of texture properties and corresponding spatial location. This chicken and egg dilemma is however a central issue in spatial segmentation applications.

The segmentation of video sequences yields the advantage that an additional feature, namely motion, can be utilized to accomplish the task. Depending on the given application, precision requirements w.r.t. the segmented objects vary. For mid-level content-based video coding, the segmented regions must not necessarily match with semantic objects. Applications enabling content-based functionalities such as object identification and retrieval require semantically meaningful objects, which are very difficult to identify generically. Motion-based segmentation typically comprises three major building blocks. The first one is the

selected region of support type, which can be pixels, regions, edges or blocks for instance. The second building block is the motion description, which can be done via a motion model [4] or optical flow estimation [4], [41]. The last component of a motion-based segmentation algorithm is the segmentation criterion, which differs from approach to approach.

The spatial and temporal segmentation algorithms developed in this thesis will be presented in the following.

4.1 Spatial Texture Analysis

In this section, a spatial texture analysis algorithm is introduced. It is required as sub-module of the spatio-temporal texture analyzer presented in Sec. 4.2.4. An adequate framework is firstly determined and then further optimized using known methods in order to achieve an efficient segmentation tool capable to provide the spatio-temporal analysis module with reliable hints.

4.1.1 Previous Work

Many segmentation methods are based either on local pixel similarities or corresponding discontinuities. Similarity or homogeneity-driven approaches encompass thresholding, clustering, region growing, as well as splitting and merging. Discontinuity or boundary-based approaches partition images based on criteria as edges. Gray value gradients are assumed to be smooth within objects and steep at object boundaries in this framework. The smoothness criterion should account for the inherent randomness of natural textures in order to avoid unrealistic homogeneity assumptions regarding within-object textures. Graph-theoretic methods also constitute an important sub-set of the boundary-based approaches. The former methods map the relation between building blocks onto a graph structure. The granularity of the building blocks is thereby algorithm-dependent and can vary from a single pixel to a texture region. Detailed descriptions of the above-mentioned approaches can be found in [42],[43],[44],[45],[46],[47],[48],[49]. Other, often application-based, classifications of segmentation strategies can be found in the literature [42],[43],[45]. The similarity vs. discontinuity categorization is preferred in this thesis, due to the existence of an in-depth quantitative

evaluation of segmentation strategies based on the previously mentioned classification [50].

Techniques exclusively relying on homogeneity or boundary criteria typically yield inaccurate segmentation results [51]. As these approaches are complementary, several hybrid algorithms have been proposed in the literature. These algorithms aim to combine the strengths of both elementary techniques to achieve improved segmentation accuracy [52],[53],[54],[55],[56],[57],[58],[59], [60],[61],[48]. Depending on the time of integration of the elementary information, hybrid approaches can be grouped into two categories: Embedded fusion and post-processing fusion [50],[62]. An overview of the preferred categorization of segmentation algorithms is given in Fig. 3.

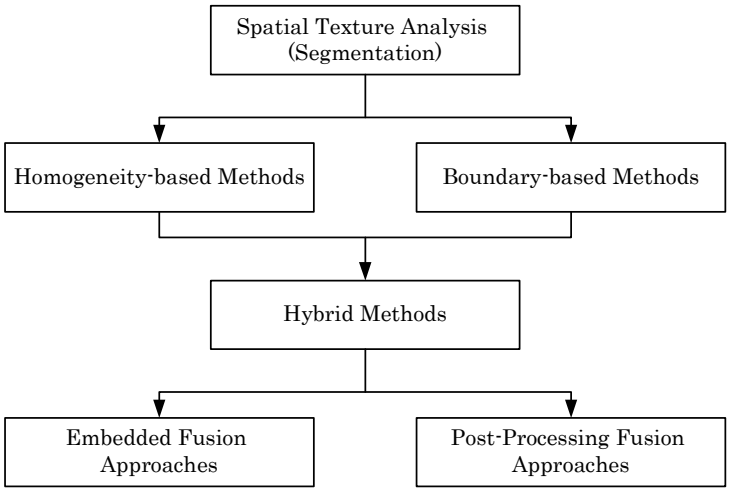


Fig. 3 – Classification of spatial segmentation algorithms

Embedded fusion algorithms are typically homogeneity-based approaches enriched with boundary analysis tools [52],[53],[54],[48]. Boundary information is often used to control the growth of homogeneous regions in split and merge or region growing approaches for instance. In similarity-driven approaches, the initial regions have to be set a priori. This is often done arbitrarily, which can yield sub-optimal segmentation results. However, constrained seed point or seed region selection can be implemented in consideration of edge information. That is, seed points or regions can be purposely placed within object boundaries to maximize the probability to achieve a good segmentation result [54].

Post-processing fusion approaches differ from embedded fusion in the sense that they carry out the integration of homogeneity and boundary information a posteriori. That is, similarity and discontinuity analysis are operated separately. The final segmentation mask is achieved by adequate combination of both segmentation results. The resulting mask is typically a refined and more accurate version of the single masks [55],[56],[57],[58],[59],[60],[61].

Freixenet et al. [50] have evaluated several representative embedded and post-processing fusion approaches. They define test conditions comprising natural and synthetic images. The segmentation results are evaluated based on objective measures introduced by Huang and Dom [63]. These measures allow a precise evaluation of boundary and region segmentation accuracy. The experimental results obtained in [50] show that, in general, post-processing algorithms yield better results than embedded approaches. Within the class of post-processing segmentation tools, the multiresolution algorithm proposed in [57] yields the best results, both in terms of segmentation accuracy and computational complexity.

In this thesis, an algorithm [56] similar to the one described in [57] is used as baseline for the required spatial texture analysis. Extensions to the framework described in [56] will be presented in the following. They aim at alleviating some of its drawbacks w.r.t. the video data that are typically used in the video coding framework addressed in this work. For better comprehension of the optimization algorithms proposed in this work, the genuine approach shall first be presented in appropriate depth.

4.1.2 Multiresolution Approach by Spann and Wilson

The randomness and uncertainty problems, that are key issues in segmentation applications as explained above, are tackled by Spann and Wilson by using a multiresolution analysis approach in [56],[57],[58]. The fundamental assumption of these approaches is the invariance of object properties across scales. This hypothesis is closely related to the work of Witkin [64] and Koenderink [65].

Fig. 4 – Segmentation algorithm by Spann and Wilson [56]

Spann and Wilson's approach [56] generates a multiresolution image pyramid by applying a quad-tree smoothing operation on the original image. Homogeneous regions are extracted at the highest level of the quad-tree, i.e. at the lowest spatial resolution, via statistical classification. The latter is based on a local centroid algorithm [66]. The classification step is followed by a coarse to fine boundary estimation based on the partition obtained at the top level of the pyramid. No a priori information, such as the number of segments, is required in this framework. The principle of Spann and Wilson's approach is depicted in Fig. 4.

4.1.2.1 Quad-Tree Smoothing

The smoothing operation is the first step of the algorithm presented in [56]. It may be operated on the original image or a transformed representation of it [56],[58]. The smoothing is performed using a conventional quad-tree approach [67],[56], which yields a hierarchy of lowpass filtered versions of the original image, such that successive ascending levels correspond to lower frequencies.

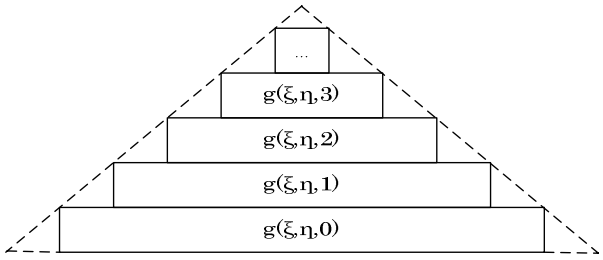


Fig. 5 – Image plane pyramid generated by quad-tree smoothing

A formal definition of the quad-tree can be given as follows. Let $g(\xi,\eta)$ stand for a 2D (transformed) image plane with $0 \leq \xi,\eta \leq N$. Note that a square image of size $N \times N$ can be assumed here without any loss of generality. It is further assumed that $N = 2^{N_{quad}}$. If this condition is not fulfilled for the given image resolution, some pre-processing may be required w.r.t. image border pixels. The

quad-tree derived from $g(\xi,\eta)$ can now be denoted $g(\xi,\eta,\ell)$, where ℓ corresponds to the considered pyramid level and $0 \le \ell \le N_{quad}$. The domain of (ξ,η) can be more generally rewritten: $0 \le \xi,\eta \le 2^{N_{quad}-\ell}$. The formal definition of $g(\xi,\eta,\ell)$ can be given as

$$g(\xi,\eta,\ell) = \frac{1}{4}g(2\xi,2\eta,\ell-1) + \frac{1}{4}g(2\xi,2\eta+1,\ell-1) + \frac{1}{4}g(2\xi+1,2\eta,\ell-1) + \frac{1}{4}g(2\xi+1,2\eta+1,\ell-1) \quad (1)$$

with

$$g(\xi,\eta,0) = g(\xi,\eta) \ . \quad (2)$$

Not only is quad-tree smoothing a fast operation, but does it also allow to balance the reduction in measurement noise via smoothing against the bias related to the fusion of information from possibly distinct homogeneous regions. The maximum smoothing gain can be obtained by truncating the quad-tree at level $\ell' < N_{quad}$, while maintaining sufficient resolution to ensure accurate segmentation of all regions whose radius r_{quad} fulfills the condition: $r_{quad} \ge 2^{\ell'+1}$. A graphical interpretation of as well as visual effects generated by quad-tree smoothing are shown in Fig. 5 and Fig. 6. Note that image planes at higher pyramid levels have been interpolated to achieve the size of the original image for better visualization of the smoothing effect in Fig. 6.

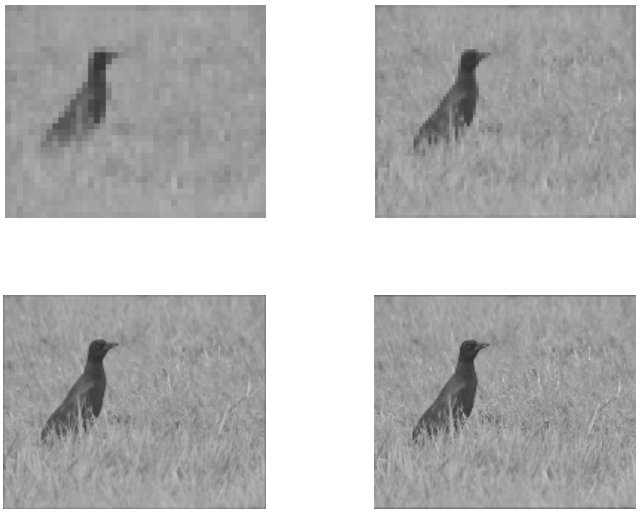


Fig. 6 – Image planes 3 (top left), 2 (top right), 1 (bottom left) and 0 (bottom right) generated by quad-tree smoothing

The quad-tree smoothing used in [56] can be easily adapted to obtain a Gaussian Pyramid [4]: A lowpass filter (Gaussian filter kernel) must be applied, at each pyramid level, before decimating the image to a quarter of its former size. This provides stronger constraints w.r.t. the sampling theorem.

4.1.2.2 Local Centroid Clustering

Local centroid clustering is applied at the highest pyramid level to achieve consistent homogeneous texture regions. The algorithm described in [66] is used. It allows a non-parametric classification of the image plane with the lowest resolution on the basis of its gray value statistics.

$$\mathbf{h}^{\tau}(u) = \sum_{u' \in W_u^{\tau}} \mathbf{h}^{\tau\text{-}1}(u') \text{ with } \mathbf{h}^{0}(u) = \mathbf{h}(u) \tag{3}$$

$$u' \in W_u^{\tau} \text{ iff } u = \text{Int } \frac{\sum_{p=-m}^{m} p\mathbf{h}(u'+p)}{\sum_{p=-m}^{m} \mathbf{h}(u'+p)} \tag{4}$$

As can be seen from (3) and (4), the local centroid algorithm moves the bin populations of the original histogram $\mathbf{h}(u)$ to their center of gravity within a sliding window W_u^{τ} of size $2m+1$. Local centroid clustering is an iterative

approach, where τ represents the number of iterations. Int() stands for the integer part of the quotient in (4).

Once the algorithm has converged, i.e. when a maximal iteration number has been reached or $\mathbf{h}^{\tau}(u) = \mathbf{h}^{\tau-1}(u)$, the classification is in principle achieved by mapping each bin of the original histogram to the corresponding center of gravity.

The implicit assumption of the local centroid method described above is that images are composed of regions with different gray level ranges, such that the corresponding histogram can be separated into a number of peaks or modes, each corresponding to one region, and there exists a threshold value corresponding to the valley between two adjacent peaks [47]. Thus the number of classes obtained through local centroid clustering depends on the "peakiness" of the original histogram $\mathbf{h}(u)$ and on the window size $2m+1$.

4.1.2.3 Boundary Estimation

The uncertainty problem described above must now be dealt with in order to robustly extract reliable region contour information from the image data. Spann and Wilson remove uncertainty by introducing a fundamental assumption: The invariance of region properties across the scales spanned by the quad-tree. Given this hypothesis, classification results obtained for image planes at higher pyramid levels can be propagated to lower levels of the tree to initialize the new classification cycle. Non-boundary pixels are assigned to the same class as their fathers, while boundary nodes are re-classified at the lower pyramid level in such a way that the boundary width is reduced by a factor of two on each step down the quad-tree (cp. Fig. 7).

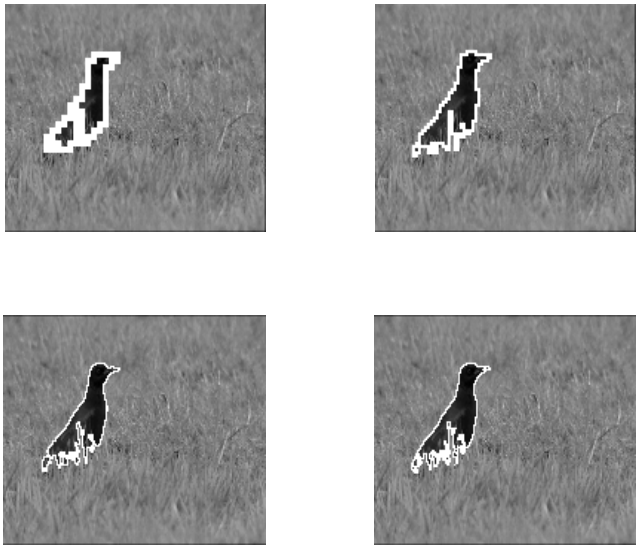


Fig. 7 – Coarse to fine boundary refinement by Spann and Wilson [56]. Image planes are depicted in the order 3 (top left), 2 (top right), 1 (bottom left), and 0 (bottom right).

A formal definition of the boundary refinement approach can be given as

$$\mathrm{cl}\left[g(\xi,\eta,\ell) \right] = \mathrm{cl}\left[g(\xi/2,\eta/2,\ell+1) \right] . \tag{5}$$

Hence, $\mathrm{cl}(g)$ is a function from IN to IN that operates the class label mapping across pyramid scales. Given the classification defined in (5), the boundary region B_ℓ^r can be defined as

$$(\xi,\eta,\ell) \in B_\ell^r \text{ iff } \mathrm{cl}\left[g(\xi,\eta,\ell) \right] \neq \mathrm{cl}\left[g(\xi',\eta',\ell) \right] \tag{6}$$

with $(\xi',\eta',\ell) \in N_8(\xi,\eta,\ell)$, where $N_8(\xi,\eta,\ell)$ is the 8-neighbor set of the pixel at location (ξ,η,ℓ) at pyramid level ℓ. B_ℓ^r is further enlarged by considering the set of nodes B_ℓ^N, which have an 8-neighbor in B_ℓ^r, defined as

$$(\xi,\eta,\ell) \in B_\ell^N \text{ iff } (\xi',\eta',\ell) \in B_\ell^r \wedge (\xi',\eta',\ell) \in N_8(\xi,\eta,\ell) \tag{7}$$

to give B_ℓ such that

$$B_\ell = B_\ell^r \cup B_\ell^N . \tag{8}$$

B_ℓ is smoothed using a linear filter whose dimension depends on the estimated signal-to-noise ratio for level ℓ. Further details about this smoothing step can be found in [56].

Fig. 8 – Original image (left) and corresponding segmentation result (right)

After smoothing, a nearest class mean classification is made on all the nodes in the enlarged boundary region B_ℓ . An example of a segmentation result is shown in Fig. 8.

4.1.2.4 Drawbacks

Histogram thresholding approaches have been first introduced for monochrome images and widely used for segmentation [68]. Applying these approaches as is on color images typically requires segmenting the luminance channel, which bears the major drawback that the multi-feature nature of color images is not exploited. That is, the homogeneous region extraction does not take advantage of the information available in the typically three color channels and thus the correlation among these components is ignored.

In this thesis, approaches have been developed that alleviate the above-mentioned drawback. They will be presented in detail in the following sections. Before doing so, color models are discussed in the next section.

4.1.3 Color Models

Color perception is one of the most important aspects of the human visual system. The latter uses color as a key discrimination criterion to resolve object identification tasks for instance. Thus the color information is a major feature in content-based image analysis applications in general and segmentation in particular [4].

Visible light represents a narrow band of the electromagnetic spectrum and is composed of waves of length 400nm (ultraviolet light) to 700nm (infrared light).

Each wavelength of this subband is perceived as an individual color tone. Wavelengths between 450nm and 500nm are perceived as blue, while they are perceived as green in the interval 500nm to 570nm, as yellow between 570nm and 590nm, and as red between 620nm and 700nm. Light composed of the same proportion of all wavelengths is perceived as white. White, black, and pure gray shades are called achromatic colors, while other colors like blue, red, green, or yellow are called chromatic. Color is perceived by humans as a combination of red (R), green (G), and blue (B) which are usually called primary colors. That is, the human eye encodes the incoming light as a pattern of dimension three, such that each color tone is represented by a characteristic activity or response pattern. This motivates the representation of digital color images with red, green, and blue components [69],[70],[4].

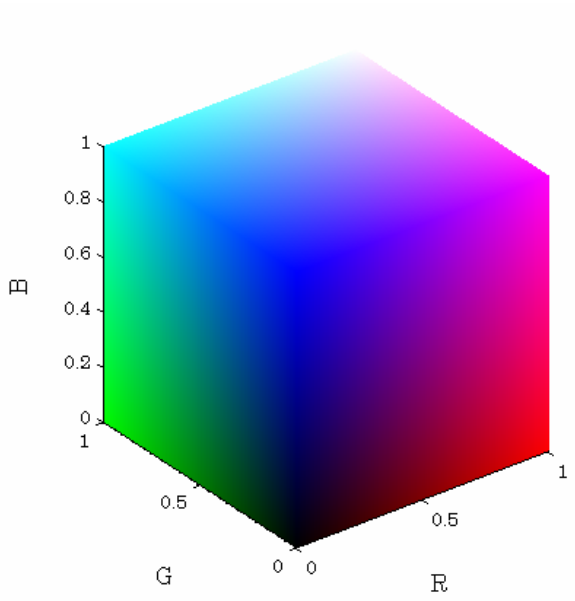


Fig. 9 – Representation of the RGB color space

The RGB color space can be represented as a cube as shown in Fig. 9. Given a normalized color representation, any color within the cube can be described by an (R,G,B) tuple, where each component of the tuple is restricted to the interval $[0\ \ 1]$. E.g. the triples $(1,0,0)$, $(0,1,0)$, and $(0,0,1)$ represent red, green, and blue respectively. The achromatic colors lie on the main diagonal of the RGB cube with the tuple $(0,0,0)$ representing black and the tuple $(1,1,1)$ representing white. Gray shades can be found between the two extremities.

Digital cameras typically output acquired pictures via the primary color components R, G, and B. Unfortunately, these features are typically not suitable for color classification due to the following reasons [4]:

- The primary color components are highly correlated, i.e. if the intensity (brightness) changes, all three components will change accordingly. These inherent redundancies may yield unnecessarily high dimensionality of the color information.
- The RGB color representation may allow color combinations which may be irrelevant in natural images
- Color differences are not represented in an uniform scale which makes it impossible to evaluate perceptual similarity from the Euclidean distance between two colors.

Given the above-mentioned disadvantages of the RGB color space, it is recommended to use statistically independent color components for color feature extraction. It is further desirable to achieve color spaces adapted to the human color perception characteristics and natural color occurrence. Hence, several color representations derived from the RGB color space have been proposed in the literature. They can be classified into linear and non-linear transformations [70], [4],[47].

Linear transformations can be represented by a matrix operation

$$\begin{pmatrix} C_1 \\ C_2 \\ C_3 \end{pmatrix} = \mathbf{T} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \tag{9}$$

where C_1, C_2, and C_3 are the color components of the target color space, while $\mathbf{T}$ represents the transformation matrix. Although the transformation in (9) is representative of diverse color spaces, note that the number of output color components can in principle be different of the number of input components. The YIQ and YUV color spaces are representatives of the class of linear transforms. YIQ and YUV are TV color spaces adapted to the American (NTSC) and European (PAL) systems respectively and characterized by their respective transformation matrices $\mathbf{T}$ (cp. (9)). In both color spaces, the color information is decomposed into a luminance component Y and two chrominance components I and Q or U and V respectively. Hence the YIQ and YUV color spaces alleviate

the correlation of the R, G, and B color components. Their transformation matrices can be found in [47].

Several non-linear color spaces are described in the literature and in use today [70],[47]: Normalized RGB, HSV, HSI, Lab, Luv, Munsell etc.. In the normalized RGB color space, colors are independent of lighting intensity variations, i.e. the latter are made uniform across the spectral distribution. The normalized color space can be formulated as

$$r = \frac{R}{R+G+B}\ , \tag{10}$$

$$g = \frac{G}{R+G+B}\ , \tag{11}$$

$$b = \frac{B}{R+G+B}\ , \tag{12}$$

with

$$r+g+b=1\ . \tag{13}$$

A drawback of this color representation is its instability at low intensities. This is due to the non-linear transform from the RGB to the normalized RGB color space. The HSI color model is based on human color perception and follows the Munsell color system [70],[47]. This model refers to hue (H), saturation (S) and Intensity (I) components. The HSI model is derived from the RGB space by the formulae

$$H = \arctan\left(\frac{\sqrt{3}(G-B)}{(R-G)+(R-B)}\right)\ , \tag{14}$$

$$I = \frac{R+G+B}{3}\ , \tag{15}$$

$$S = 1-\frac{\min(R,G,B)}{I}\ , \tag{16}$$

where H represents an angle as can be seen in (14). Color information is separated from intensity information in the HSI model. The former is represented by hue and saturation. Hue is defined as the basic color of light and can be determined by the dominant wavelength in the corresponding spectral

distribution of wavelengths. Saturation corresponds to the relative amount of white light in a chromatic color and is inversely proportional to that amount.

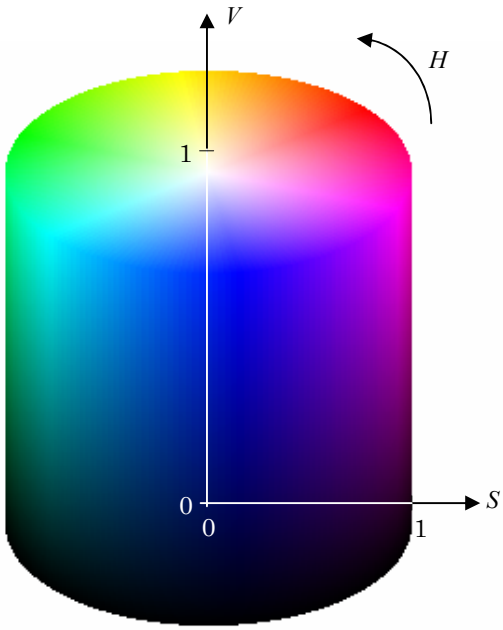


Fig. 10 – Illustration of the HSV color space

The intensity component describes the brightness of an image and is determined by the amount of the light. The HSI model features singularities at $S = 0$ and $I = 0$. Note that $I = 0$ if and only if $R = G = B = 0$. Thus the hue values near the singularity are numerically unstable. This is a problem that has to be tackled when using HSI. The HSV color space is a variant of HSI. The difference between the two is that V is determined as the maximum of the R, G, and B components. An illustration of the HSV color space can be found in Fig. 10. The Luv and Lab color spaces are perceptually uniform and can be indirectly, non-linearly derived from the RGB color space. They provide better separation of color from luminance (intensity) information than the RGB model and are particularly efficient for measuring small color differences. However, singularity problems are given here too.

4.1.4 Histogram Thresholding via Color Channel Pruning

The experiments conducted by Freixenet et al. [50] show that Spann and Wilson's [56] segmentation approach yields better results than other segmentation paradigms. Thus the approach presented in [56] and sketched in Sec. 4.1.2 is used as baseline for further improvements w.r.t. color image segmentation.

4.1.4.1 Previous Work

Several attempts to extend histogram clustering approaches to color images have been made in the past. The fundamental difference between monochrome and color images is that the latter are represented by a tri-stimulus R, G, and B or their linear, respectively non-linear, transformations. Thus determining appropriate thresholds in this multidimentional space is reasonably complex. The approaches presented in [71],[72],[73] reduce this three and more dimensional problem to a one dimensional task. This is done by recursively switching between the available color channels, i.e. the corresponding histograms, such that the histogram with the best peak is selected to carry out the given segmentation task.

4.1.4.2 The Algorithm

The fundamental histogram clustering principle described in [71],[72],[73] can be easily integrated into the framework by Spann and Wilson. For that, a channel pruning unit is inserted into the processing chain as shown in Fig. 11.

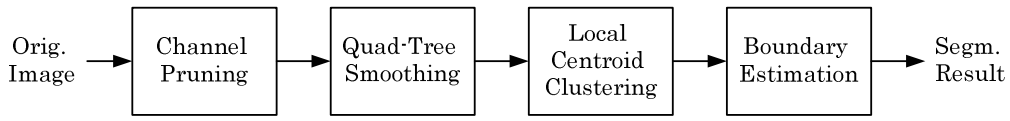


Fig. 11 – Schematic representation of the segmentation algorithm by Spann and Wilson [56] with an additional color channel pruning unit

The channel pruning step consists in selecting a color channel for multiresolution segmentation, such that the within-cluster variance is minimized and the between-cluster distance is maximized. The formal definition of the cost function used in this thesis is given by

$$E = -\frac{N_C(N_C - 1)}{2} \quad \frac{\sum_{\mu_n \in C} \sum_{\substack{\mu_p \in C - \{\mu_n\} \\ \mu_p > \mu_n}} \left\| \mu_n - \mu_p \right\|}{\sum_n \sigma_n^2} \quad \text{with} \quad 0 \le n,p \le N_C \tag{17}$$

where C represents the set of cluster centroids sorted in ascending order. N_C is assumed to be the number of cluster centroids in C. Note that the centers of gravity correspond to bin locations within a given histogram. The quotient depending on N_C is a normalization factor to the numerator (sum of inter-cluster distances) of the second quotient. Note that the norm used to determine the distance between two clusters can be chosen according to application requirements. However, the ℓ_1 norm is used in this thesis. σ_n^2 is the within-cluster variance of the cluster represented by the centroid μ_n. The histogram of the input color channel that minimizes the cost function is used for multiresolution segmentation, as it potentially yields the best result given the adopted segmentation strategy.

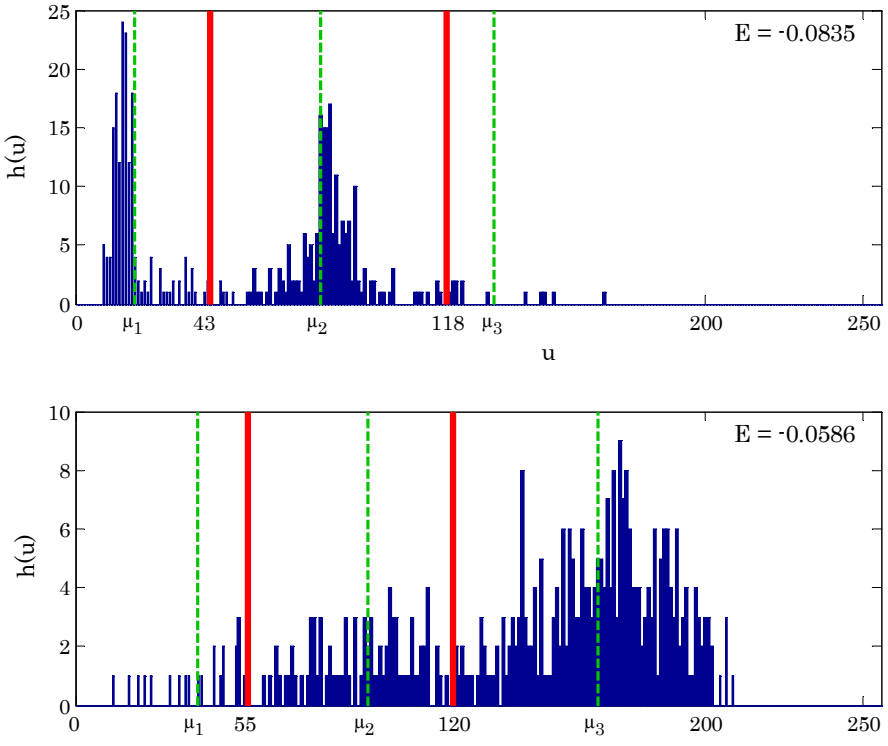


Fig. 12 – Principle of the channel pruning approach

Two histograms are exemplarily depicted in Fig. 12. They correspond to different color channels of the same image. In fact, the image considered here shows an object in front of a homogeneous background of a different color. It can be easily seen that the top histogram features better modes than the second. The costs are consequently smaller in the former than in the latter case. The cluster centroids determined by the local centroid clustering algorithm [66] are represented by μ_1, μ_2, and μ_3 in Fig. 12. The clusters are delimited by red bars. For instance, the three clusters found in the top histogram are $\{0,\ldots,43\}$, $\{44,\ldots,118\}$, and $\{119,\ldots,256\}$.

Channel discrimination is done at the highest pyramid level, i.e. at the lowest spatial resolution, in order to achieve robust homogeneous regions. The required cluster centroids as well as limits are extracted using the local centroid clustering algorithm described in [66] and used by Spann and Wilson [56]. Thus in the strict sense, the quad-tree and local centroid clustering modules can be seen as sub-modules of the channel pruning step.

4.1.5 Histogram Thresholding via Redundancy Elimination

Although the approach presented in Sec. 4.1.4 provides automatic selection of the best of the available color channels for the segmentation task, it does not take the possible correlation between the color components into account. Hence the segmented regions are based on just one of at least three color features.

4.1.5.1 Previous Work

Approaches alleviating this drawback have been presented in [74],[75],[76]. The approach by Celenk [74] is an attempt to achieve an optimal linear dimensionality reduction. The points in the 3-dimensional space are projected onto a line, such that they can ideally be well separated. The line is estimated using Fisher's linear discriminant method [77]. In general, the projection onto one dimension yields significant loss of information, such that classes that are well separated in the original 3-dimensional space become strongly overlapping in one dimension. However, cluster separation can be maximized by adjusting Fisher's parameters [77]. The approaches by Ohta et al. [75] and Tominaga [76] are similar to each other in the sense that they extract color features with high discrimination power using principal component analysis (PCA) [77]. PCA is an

approach that is typically used for linear dimensionality reduction. In contrast to Fisher's linear discriminant method, PCA inherently allows for better, i.e. explicit, control of the error introduced by dimensionality reduction. The approach introduced in [75] is further considered in this thesis, as it is the genuine publication.

4.1.5.2 The Algorithm

Consider a region R in a d-dimensional color space. RGB will be considered in the following ($d = 3$), but other color spaces (e.g. HSV) may also be used. Let $\mathbf{h}_R$, $\mathbf{h}_G$, and $\mathbf{h}_B$ stand for the distributions of R, G, and B in R respectively. Assume that $\mathbf{f}$ is a row of the matrix $[\mathbf{h}_R\, \mathbf{h}_G\, \mathbf{h}_B]$ of dimension $N_{hist} \times 3$, where N_{hist} is the histogram resolution. Then the vector $\mathbf{f}$ can be represented, without any loss of generality, as a linear combination of orthonormal vectors $\mathbf{f}_p$

$$\mathbf{f} = \sum_{p=1}^{d} z_p \mathbf{f}_p \tag{18}$$

where z_p are the coefficients of the linear combination. The vectors $\mathbf{f}_p$ satisfy the orthonormality condition

$$\mathbf{f}_p^{\mathrm{T}} \mathbf{f}_n = \begin{cases} 1\,, & p = n \\ 0, & \text{else} \end{cases}\,. \tag{19}$$

The coefficients z_p can be found in consideration of (19) to give

$$z_p = \mathbf{f}_p^{\mathrm{T}} \mathbf{f} \tag{20}$$

which can be seen as a simple rotation of the original coordinate system towards the new set of coordinates given by the z_p. Further details about this interpretation can be found in [77].

Suppose that only a subset $M_{hist} < d$ of the basis vectors $\mathbf{f}_p$ is retained. Then each vector $\mathbf{f}$ is approximated by an expression of the form

$$\tilde{\mathbf{f}} = \sum_{p=1}^{M_{hist}} z_p \mathbf{f}_p + \sum_{p=1+M_{hist}}^{d} b_p \mathbf{f}_p \tag{21}$$

where b_p are constants. $\tilde{\mathbf{f}}$ thus has M_{hist} degrees of freedom in contrast to $\mathbf{f}$ which has d of them. Now consider the set of N_{hist} row vectors $\{\mathbf{f}_u\}$ ($u = 1, 2, \ldots, N_{hist}$) of the matrix $[\mathbf{h}_R \, \mathbf{h}_G \, \mathbf{h}_B]$.

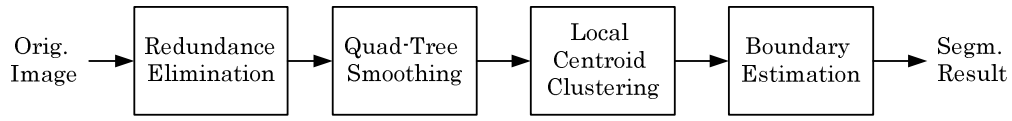


Fig. 13 – Segmentation algorithm by Spann and Wilson [56] with an additional color channel redundancy elimination unit

The best approximation of the original vector $\mathbf{f}$ on average for the whole data set can then be achieved by minimizing the sum of square errors

$$\mathrm{E} = \frac{1}{2} \sum_{p=1+M_{hist}}^{d} \mathbf{f}_p^{\mathrm{T}} \boldsymbol{\Sigma} \mathbf{f}_p \tag{22}$$

where $\boldsymbol{\Sigma}$ represents the covariance matrix of the set of vectors $\{\mathbf{f}_u\}$ and is given by

$$\boldsymbol{\Sigma} = \sum_{u} (\mathbf{f}_u - \bar{\mathbf{f}})(\mathbf{f}_u - \bar{\mathbf{f}})^{\mathrm{T}} \tag{23}$$

where $\bar{\mathbf{f}}$ is the mean vector defined as

$$\bar{\mathbf{f}} = \frac{1}{N_{hist}} \sum_{u=1}^{N_{hist}} \mathbf{f}_u \ . \tag{24}$$

It can be shown that the minimum occurs when the basis vectors satisfy

$$\boldsymbol{\Sigma} \mathbf{f}_p = \lambda_p \mathbf{f}_p \tag{25}$$

such that they are the eigenvectors of the covariance matrix [77]. Substituting (25) into (22), and making use of the orthonormality relation in (19), the value of the error criterion at the minimum is obtained in the form

$$\mathrm{E} = \frac{1}{2} \sum_{p=1+M_{hist}}^{d} \lambda_p \quad . \tag{26}$$

The minimum error is obtained by choosing the $d - M_{hist}$ smallest eigenvalues and their corresponding eigenvectors as the ones to discard.

Assume that $\lambda_1 \geq \lambda_2 \geq \lambda_3$ $(d = 3)$ and that the coordinates of the eigenvectors of $\boldsymbol{\Sigma}$ are given by $\mathbf{f}_p = (f_p^R, f_p^G, f_p^B)$ (RGB color space), then the color features F_1, F_2, and F_3 are defined as

$$F_p = \begin{pmatrix} f_p^R \\ f_p^G \\ f_p^B \end{pmatrix}^{\mathrm{T}} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \text{ with } p = 1,2,3 \quad . \tag{27}$$

where R, G, and B are the color components of a given sample in a given input image. The sample coordinates are ignored here for better legibility. It can be shown that F_1, F_2, and F_3 are uncorrelated and that F_1 has the largest variance equal to λ_1, and thus has the largest discriminant power. F_2 has the largest discriminant power among the vectors orthogonal to F_1 [77]. In this thesis, F_1 is used for multiresolution segmentation.

4.2 Spatio-Temporal Texture Analysis

4.2.1 Fundamentals

In this section, insight is given into fundamental motion-related tools for spatio-temporal video analysis. For better legibility, the continuous spatial coordinates (x, y) are used throughout the following sections, although, in some cases, they may relate to discrete coordinates.

4.2.1.1 Parametric Motion Models

Motion in a video sequence is typically due to camera operations or moving objects. Both motion sources can be analyzed using 2D parametric motion models. Several of the latter can be found in the literature [4]. They differ from each other with regard to the complexity of the motion they can model and further, more or less restrictive inherent assumptions. They all have the advantage of being compact, i.e. the motion of a given region can be completely

described with just a few model parameters. Common parametric motion models are the translational motion model given as

$$\begin{aligned} v_x &= x' - x = a_1 \\ v_y &= y' - y = a_2 \end{aligned}, \tag{28}$$

the affine motion model defined as

$$\begin{aligned} v_x &= a_1 + a_3 x + a_4 y - x \\ v_y &= a_2 + a_5 x + a_6 y - y \end{aligned}, \tag{29}$$

and the perspective motion model given as

$$\begin{aligned} v_x &= [(a_1 + a_3 x + a_4 y)/(1 + a_7 x + a_8 y)] - x \\ v_y &= [(a_2 + a_5 x + a_6 y)/(1 + a_7 x + a_8 y)] - y \end{aligned}, \tag{30}$$

where sample coordinates (x, y) in a given picture are warped towards (x', y') in another picture. $v_{x,y}$ correspond to the horizontal and vertical velocities supposing a Cartesian coordinate system. $(a_1, \ldots, a_8)$ is the motion parameter vector, with $a_{1,2}$ being translation parameters, $a_{3,6}$ being scaling parameters, $a_{4,5}$ being shearing parameters, and $a_{7,8}$ representing perspective motion parameters. Note that this interpretation of the physical significance of the model parameters holds only if the origin of the considered coordinate system resides at the centroid of the considered object in the reference image, i.e. in the image that is warped towards the current one [78]. All motion parameters are real numbers. Other parametric motion models can be found in the literature. The reader is referred to [4] for an in-depth discussion of these.

The perspective motion model is non-linear and the most complex of the three models defined above, due the fact that it possesses the largest number of parameters. It approximates object surfaces with plane surfaces, which holds only if depth discontinuities within a rigid object are negligible compared to the distance between object and image plane of the camera. Given this simplification, $v_{x,y}$ can be formulated independently of the camera's focal length and the object's distance to the camera. The perspective motion model is suitable to describe arbitrary rigid object motion, if the camera operation is restricted to pure

rotation and zoom. It is also suitable for rigid motion of planar objects with arbitrary camera operation.

The affine and the translational motion models are less complex than their perspective counterpart is. They are linear transforms and thus mathematically easier to manage. On the other hand, they exhibit harder constraints than the perspective motion model. In fact, they assume orthographic projection, i.e. they presume that the considered object is far away from the image plane of the camera. For that, the focal length of the camera and the depth are assumed to be constant, which a priori yields a less accurate motion description. The orthographic projection assumption is violated in case of large 3D objects with significant depth discontinuities near the camera and translation of the same object perpendicularly to the image plane of the camera.

All the motion models presented here lack parameters to model distortions caused by the camera's lens. For that, effects due to this phenomenon are explicitly ignored.

4.2.1.2 Motion Estimation

Once a motion model has been selected for a given application, an algorithm for the corresponding parameter estimation must be implemented. Several motion estimation methods have been presented in the literature [4],[78]. In the framework of the spatio-temporal texture analyzer developed in this thesis, a motion estimation approach presented in [78] is used.

Let's convey the perspective motion parameters into a vector as

$$\mathbf{p} = \left(a_1, a_3, a_4, a_2, a_5, a_6, a_7, a_8\right)^{\mathrm{T}} \quad . \tag{31}$$

Let's further consider N_{ref} samples $\left(x^{(n)}, y^{(n)}\right)$ in the reference image, with $n = 1,..., N_{ref}$. The motion is to be jointly estimated at these locations. Ideally, the N_{ref} samples would be part of the same object. Considering a temporally shifted image, called the current image in the following, let's assume that the samples $\left(x^{(n)}, y^{(n)}\right)$ have been displaced towards $\left(x^{(n)'}, y^{(n)'}\right)$ in the current picture.

Conveying the sample coordinates $\left(x^{(n)'},y^{(n)'}\right)$ into a so-called measurement vector of size $2N_{ref}$ x1 yields

$$\mathbf{k}=\begin{pmatrix} x^{(1)'} \\ y^{(1)'} \\ \vdots \\ x^{(N_{ref})'} \\ y^{(N_{ref})'} \end{pmatrix}. \tag{32}$$

The functional relationship between (31) and (32) can now be established under consideration of (30) as follows

$$\mathbf{k}=\mathbf{D}\mathbf{p}+\boldsymbol{\eta} \tag{33}$$

with $\boldsymbol{\eta}$ being the measurement and modeling error vector and $\mathbf{D}$ being a matrix of size $2N_{ref}$ x8 defined as

$$\mathbf{D}=\begin{pmatrix} 1 & x^{(1)} & y^{(1)} & 0 & 0 & 0 & -x^{(1)}x^{(1)'} & -y^{(1)}x^{(1)'} \\ 0 & 0 & 0 & 1 & x^{(1)} & y^{(1)} & -x^{(1)}y^{(1)'} & -y^{(1)}y^{(1)'} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x^{(N_{ref})} & y^{(N_{ref})} & 0 & 0 & 0 & -x^{(N_{ref})}x^{(N_{ref})'} & -y^{(N_{ref})}x^{(N_{ref})'} \\ 0 & 0 & 0 & 1 & x^{(N_{ref})} & y^{(N_{ref})} & -x^{(N_{ref})}y^{(N_{ref})'} & -y^{(N_{ref})}y^{(N_{ref})'} \end{pmatrix}. \tag{34}$$

As can be seen in (34), $\mathbf{D}$ is dependent on the data. Ignoring $\boldsymbol{\eta}$ yields the estimation error vector

$$\mathbf{e}=\mathbf{k}-\mathbf{D}\mathbf{p} \tag{35}$$

where the prediction error vector $\mathbf{e}$ is typically minimized using an adequate error norm as

$$\|\mathbf{e}\|_{\min}=\|\mathbf{k}-\mathbf{D}\mathbf{p}\|_{\min}. \tag{36}$$

Using the Euclidian norm, the minimum is determined by setting the partial derivatives of (36), w.r.t. the motion parameters $\mathbf{p}$, to zero. This gives the following system of equations [78]

$$\mathbf{p}=\left(\mathbf{D}^{\mathrm{T}}\mathbf{D}\right)^{-1}\mathbf{D}^{\mathrm{T}}\mathbf{k}. \tag{37}$$

That is, the motion parameters can be directly determined from the equation above in case the so-called Gaussian normal matrix $\mathbf{D}^T\mathbf{D}$ is invertible. Note that $(\mathbf{D}^T\mathbf{D})^{-1}\mathbf{D}^T$ is referred to as the pseudo-inverse of $\mathbf{D}$.

It should be noticed that the displacement vectors $\left(x^{(n)'} - x^{(n)}, y^{(n)'} - y^{(n)}\right)$ also called motion field are assumed to be known in (37). They can, for instance, be estimated using optical flow estimation techniques as will be explained in the following section. (37) also requires the consideration of a neighborhood and implicitly assumes a single motion scenario. In fact, in practice, the considered neighborhood may reveal multiple motions. In that case, no single motion parameter set exists that could properly fit the motion properties of the entire neighborhood. Hence must the dominant motion be identified and the remainder motions ignored in such cases. This can be achieved by "robustifying" (37) through robust statistics as maximum likelihood estimation. Robust motion estimators will be discussed in the following sections.

A thorough discussion and classification of existing motion estimation algorithms is beyond the scope of this thesis. For an in-depth description of motion estimation algorithms not discussed in this work, the reader is referred to [4],[78].

4.2.1.3 Optical Flow Estimation

Fundamentals

Optical flow approaches correspond to motion estimation techniques based on so-called motion vector fields. These methods are rooted in hydrodynamics, where they are used to describe the motion of fluids. In image processing, the density of the optical flow field is scaled by defining a grid that can be pixel-accurate or block-accurate (set of samples are merged to a so-called block). A displacement vector is estimated for each grid element. The former describes to motion of the latter between two images. The set of motion vectors at a picture transition is called motion vector field. In the event of a pixel-accurate grid, the optical flow field is referred to as dense. Optical flow estimation approaches are typically constrained by data conservation and spatial coherence assumptions.

The data conservation constraint stands for the assumption that, given motion, the luminance remains unchanged in a differential time interval dt at the differential location (dx, dy). That is, the luminance of a region is assumed to remain constant, while its location may change. The data conservation assumption can formally be written as [4]

$$\mathrm{I}(x, y, t) = \mathrm{I}(x + dx, y + dy, t + dt) \tag{38}$$

where $\mathrm{I}(x, y, t)$ represents the luminance channel. The so-called optical flow equation is achieved by applying Taylor series approximation on (38) as follows

$$\mathrm{I}(x + dx, y + dy, t + dt) = \mathrm{I}(x, y, t) + \frac{\partial \mathrm{I}}{\partial x} dx + \frac{\partial \mathrm{I}}{\partial y} dy + \frac{\partial \mathrm{I}}{\partial t} dt + \varepsilon \tag{39}$$

where ε corresponds to a non-linear residual term. Ignoring ε yields

$$\frac{\partial \mathrm{I}}{\partial x} dx + \frac{\partial \mathrm{I}}{\partial y} dy + \frac{\partial \mathrm{I}}{\partial t} dt = 0 \tag{40}$$

which can be rewritten as

$$\frac{\partial \mathrm{I}}{\partial x} \frac{dx}{dt} + \frac{\partial \mathrm{I}}{\partial y} \frac{dy}{dt} + \frac{\partial \mathrm{I}}{\partial t} = 0 \tag{41}$$

or

$$\mathrm{I}_x v_x + \mathrm{I}_y v_y + \mathrm{I}_t = 0 \tag{42}$$

where the definition of the terms $\mathrm{I}_{x,y,t}$ and $v_{x,y}$ can be achieved through matching of (42) and (41) as

$$\mathrm{I}_x = \frac{\partial \mathrm{I}}{\partial x}, \mathrm{I}_y = \frac{\partial \mathrm{I}}{\partial y}, \mathrm{I}_t = \frac{\partial \mathrm{I}}{\partial t}, \; v_x = \frac{dx}{dt}, v_y = \frac{dy}{dt} \; . \tag{43}$$

(42) is called the optical flow equation, while $v_{x,y}$ are referred to as optical flow or horizontal and vertical velocities. Gradient-based methods assume the optical flow to be almost constant within a neighborhood N. For that, the data constraint based cost function can be derived from (42) as [41]

$$\mathrm{E}_\mathrm{D} = \sum_{(x,y) \in N} \left(\mathrm{I}_x v_x + \mathrm{I}_y v_y + \mathrm{I}_t \right)^2 \; . \tag{44}$$

Regression methods use the same data constraint formulation as given in (44). They yet allow for consideration of larger neighborhoods by modeling the local motion field using a parametric motion model [41]. In order to determine the two unknown $v_{x,y}$ uniquely from (42) or (44), at least one additional constraint must be formulated.

As the data constraint, the spatial coherence constraint assumes a single motion within a given neighborhood, which corresponds to the assumption of continuous flow variations within the neighborhood. Several mathematical formulations of the spatial coherence constraint have been proposed in the literature. One of the most usual formulations can be given as [41]

$$\mathrm{E_S} = \left(\frac{\partial v_x}{\partial x}\right)^2 + \left(\frac{\partial v_x}{\partial y}\right)^2 + \left(\frac{\partial v_y}{\partial x}\right)^2 + \left(\frac{\partial v_y}{\partial y}\right)^2 \tag{45}$$

or

$$\mathrm{E_S} = \sum_{R \subset S_R} \left[\frac{1}{8} \sum_{R' \subset N_R} \left[(v_x^R - v_x^{R'})^2 + (v_y^R - v_y^{R'})^2 \right] \right] \tag{46}$$

where R corresponds to a rectangular block of the set S_R and N_R to the set of north, south, east, and west neighbors. (45), also called first-order or membrane model [41], is minimized within a defined neighborhood as well as (46).

The assumptions made by the data and spatial coherence constraints are often violated due to the fact that the optical flow in a scene is typically piecewise smooth. Fairly common effects as specular reflections, transparency, occlusion and depth discontinuities within a rigid object equally infringe upon the data and spatial coherence assumptions.

The optical flow estimation has been extensively discussed in the literature [41],[79],[80],[81],[82]. In this thesis, the algorithm proposed by Black and Anandan [41] will be used due to its robustness to multiple motions, transparency, occlusions and specular reflections. The principle of the approach in [41] will be presented in the following.

Black and Anandan's Approach

The formulation of the data constraint in (44) entails the necessity to define the size of the neighborhood N. A large neighborhood is desirable in order to adequately constrain the solution and provide insensitivity to noise. Then again, a tight neighborhood is preferable to avoid violating assumptions with regard to single motion in the considered region. Black and Anandan refer to this chicken and egg dilemma as the generalized aperture problem.

In the event of multiple motions in the considered neighborhood, no single optical flow field can be determined that fits all the real displacements and thus yields small residuals over the whole neighborhood. For that, from the view point of one of the multiple motions, the other motions can be interpreted as statistical outliers. Least square formulations as (44) are precisely very sensitive to outliers and typically yield poor results in a noisy context.

Hence, Black and Anandan propose a robust reformulation of the spatial coherence and data constraints. Robustness thereby refers to the insensitivity of the motion estimator to assumption violations. They show that the incorporation of robust estimation techniques into the optical flow computation framework yields significant improvements w.r.t. multiple motions, transparency, occlusions and specular reflections, which are challenging issues in this context. The authors apply their framework to a number of standard approaches as correlation, area-based correlation, correlation with regularization and gradient-based approaches with regularization [41]. The robust formulation of the gradient-based approach with regularization, used in this thesis, can be given as

$$\mathrm{E} = \lambda_D \, \mathrm{E_D} + \mathrm{E_S} = \lambda_D \, \rho_{\mathrm{D}}(\mathrm{I}_x \, v_x^R + \mathrm{I}_y \, v_y^R + \mathrm{I}_t, \theta_D) + \sum_{R' \subset N_R} \left[\rho_{\mathrm{S}}(v_x^R - v_x^{R'}, \theta_S) + \rho_{\mathrm{S}}(v_y^R - v_y^{R'}, \theta_S) \right] \tag{47}$$

where $\rho_{\mathrm{D}}(x, \theta_D)$ and $\rho_{\mathrm{S}}(x, \theta_S)$ correspond to error norms insensitive to outliers. θ_D and θ_S represent scale parameters. Parameter λ_D controls the relative influence of each of the cost functions in (47). In principle, different measures could be used for the data and the spatial coherence constraints. Note that the first term of the cost function above, i.e. the regularization term, corresponds to

(44) while the second term corresponds to (46). In [41], the Lorentzian error norm is used for both constraint terms. Fig. 14 depicts the Lorentzian and the least square (Euclidian Distance) functions in its first column. The first derivative $\psi(x,\theta)$ of both is shown in the second column. $\psi(x,\theta)$ is proportional to the so-called influence function [41] that captures the bias of a particular measurement x on the solution. In Fig. 14, it can be seen that large measurements, which are typical for outliers, yield a large bias in the case of the Euclidian distance. The influence of the measurements increases linearly and without bounds. In contrast, the Lorentzian error norm features a redescending influence function, i.e. the influence of large measurements decreases beyond a threshold that is steered by the parameter θ. Applying the Lorentzian error norm to (47) yields a framework that is more robust w.r.t. to assumption violations than the least squares approach. Other error norms with similar properties as the Lorentzian norm are presented in [41].

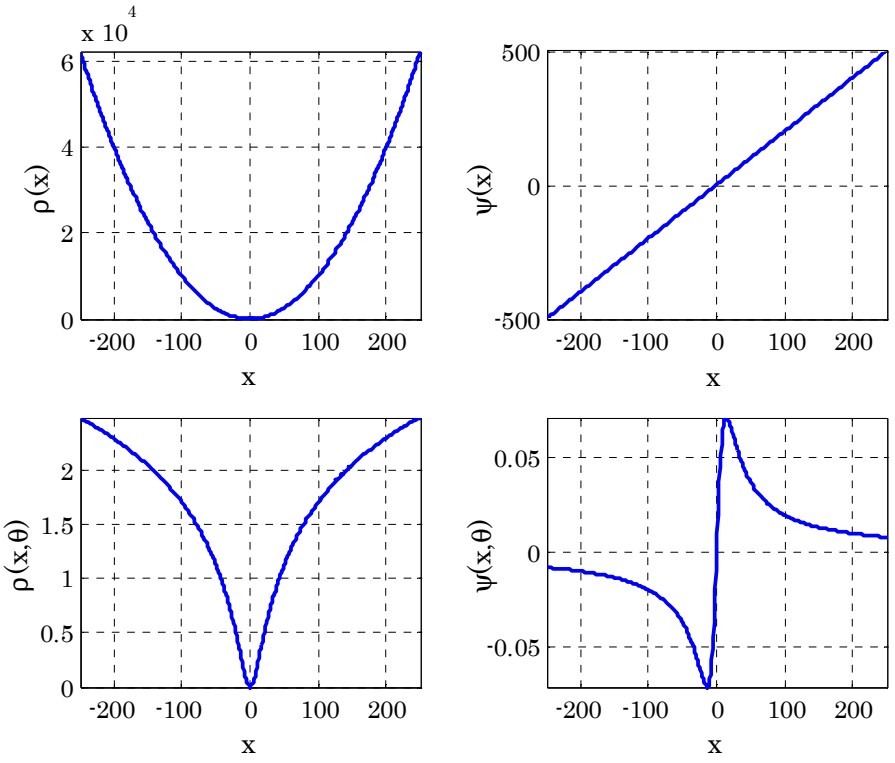


Fig. 14 – Least Square error norm (top left), Lorentzian error norm (bottom left), and corresponding influence functions (top right and bottom right respectively), Source: [41]

(47) is minimized using a deterministic continuation method described in [41]. Note that Black and Anandan apply a multiresolution strategy to identify motions larger than one pixel. For that, a Gaussian pyramid is built and, in a coarse to fine process, the optical flow is propagated from the top to the bottom of the pyramid. That is, the optical flow estimation in a given image plane is initialized by the optical flow obtained in the adjacent, lower scaled plane. The initial optical flow is then refined in the current plane.

4.2.2 Previous Work

Several segmentation algorithms for image sequences have been proposed in the literature. They can be characterized by the nature of the features they use to solve the segmentation task. Some approaches exclusively rely on motion or spatial information, while other, more generic algorithms combine motion and spatial attributes. In the following, these segmentation strategies will be referred to as motion-based (temporal), spatial and spatio-temporal respectively. The block diagram depicted in Fig. 15 shows a classification of video signal segmentation algorithms. Note that shot boundary detection approaches are not considered here, as only object segmentation algorithms are addressed in this section.

It can be seen in Fig. 15 that the spatial segmentation methods used for video segmentation are derived from still image approaches (cp. Fig. 3). Spatially-driven video partitioning algorithms typically feature an additional temporal tracking step based on spatial coherence criteria as color or texture. A number of such approaches can be found in the literature [83],[84],[85].

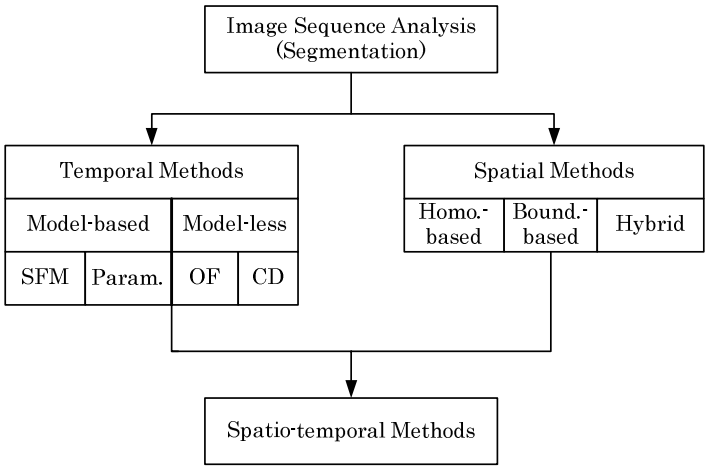


Fig. 15 – Classification of image sequence segmentation algorithms

Motion-based approaches can be subdivided into model-based and model-less algorithms (cp. Fig. 15). Model-less approaches typically analyze the luminance channel for segmentation without prior assumptions regarding object properties. These algorithms are either based on Change Detection (CD) or optical flow discontinuity analysis. Optical Flow (OF) computation corresponds to a motion estimation principle that constitutes a research area in its own right [4],[41]. Descriptions of relevant algorithms based on the segmentation of the optical flow field can be found in [86],[87],[88],[89],[90],[91],[92],[93],[94],[95]. Change detection consists in separating static background from foreground objects given an image sequence. Subsequent object segmentation and tracking can then be restricted to the areas marked as "changed" in the considered image transitions. Change detection algorithms based on simple differencing and thresholding [96], [97],[99],[100],[102],[103],[16], significance and hypothesis tests [101],[104],[105], [98],[106], predictive models [107],[108],[109],[110], or shading models [111],[112],[113],[114],[98],[106] have been proposed in the literature. An extensive survey on change detection methods can be found in [115].

In contrast to model-less algorithms, model-based methods make assumptions about the geometric modifications a given moving object undergoes with time as well as on the corresponding surface properties. Model-based approaches can be split into Structure From Motion (SFM) [116],[117],[118] and parametric [119], [120],[121],[15],[122],[123],[124],[125],[126],[127],[96],[97],[12] methods. SFM and parametric algorithms differ in their underlying motion hypothesis. While

the former typically assume rigid object motion, the latter relax this constraint to piecewise rigid motion. SFM methods aim to recover 3D geometry in space from 2D motion in the image plane. For that, 3D scenes with significant depth information are typically required. Most SFM techniques exploit geometric and algebraic characteristics that are invariant under projection onto multiple 2D views of a given scene [128]. No depth information is assumed in the parametric segmentation context. Homogeneously moving segments are described using parametric motion schemes like the affine or the perspective model (cp. Sec. 4.2.1). The latter are typically used to piecewise fit the optical flow field in order to extract rigid motion areas [119],[120],[121],[122],[15],[123],[124]. Some parametric algorithms alternatively apply the models to given regions for motion estimation by matching [4],[125],[126],[127]. Segmentation algorithms based on both model-based and model-less techniques, e.g. [96],[97],[12],[127], have also been proposed in the literature. Purely motion-based algorithms are typically noise-sensitive, inaccurate, and liable to over-segmentation.

To overcome the respective drawbacks of motion-based and spatial segmentation methods, spatio-temporal approaches have been proposed. In this framework, spatial information is often used to constrain the temporal segmentation results [97],[98],[129],[130],[123],[125],[126],[127],[124],[106],[131],[16]. The latter are typically based on short-term motion estimation. Motion similarity is thereby evaluated by an adequate norm in the motion parameter space or in the spatial domain as already explained above. Alternative methods do not require an arbitrary prioritization and thus a sequential processing of spatial and temporal features. They rather handle the latter features simultaneously [132],[133],[134],[135]. It should be noticed that some authors conduct spatio-temporal video segmentation in the transform domain. Zhu et al. [136] for instance use DCT-based features to exploit spatio-temporal correlations in video sequences.

The European research forum, COST 211, has developed an Analysis Model (AM) for video object segmentation. This forum aims to facilitate the creation and maintenance of a high level expertise in the area of video compression and related application fields in Europe. The COST 211quat project focused on the investigation of algorithms for image sequence analysis in order to increase the

acceptance of content-based functionalities provided by MPEG-4 and MPEG-7. The project aimed at supplying multimedia applications with tools that enable better exploitation of aforesaid functionalities [106],[131]. The COST 211quat project, that started in May 1998, was carried out until 2003 by a couple of European research labs and companies, which (at least some of them) significantly contributed to the standardization process of MPEG-4. For that, the COST 211quat AM will be used as anchor in this work and shall be presented in detail in the following section.

The author is aware of the fact that the QIMERA project [137] is the most recent European effort to develop a modular segmentation and tracking algorithm. The latter can not be considered in this thesis as the analysis model of the above-mentioned project is not publicly available.

4.2.3 COST 211quat AM

The description of the COST 211quat AM provided in this section is based upon a document released by the Algorithm Subgroup of the project [106] and a publicly available publication [131]. The COST 211quat AM will be called the AM in the following for simplicity matters.

The AM is a spatio-temporal framework for the analysis of video sequences $g(\xi,\eta,\ell)$. A simplified diagram depicting the principle of the AM is shown in Fig. 16. The AM incorporates color information for spatial analysis as well as change detection and parametric object motion description for temporal analysis. Note that these temporal analysis modules are represented by the Motion Analysis block in Fig. 16. The analysis units Partition Tracking and Local Object Feature Extraction purvey additional hints that can be used to refine the segmentation. The segmentation masks resulting from the above-mentioned analysis units are merged, by a so-called rule processor, in a post-processing step to obtain the final segmentation result $m(\xi,\eta,t)$. Pre-processing consists in temporal alignment of the input pictures and scene cut detection. Temporal alignment is done to improve the local motion analysis, while scene cut detection is used to detect significant video content changes.

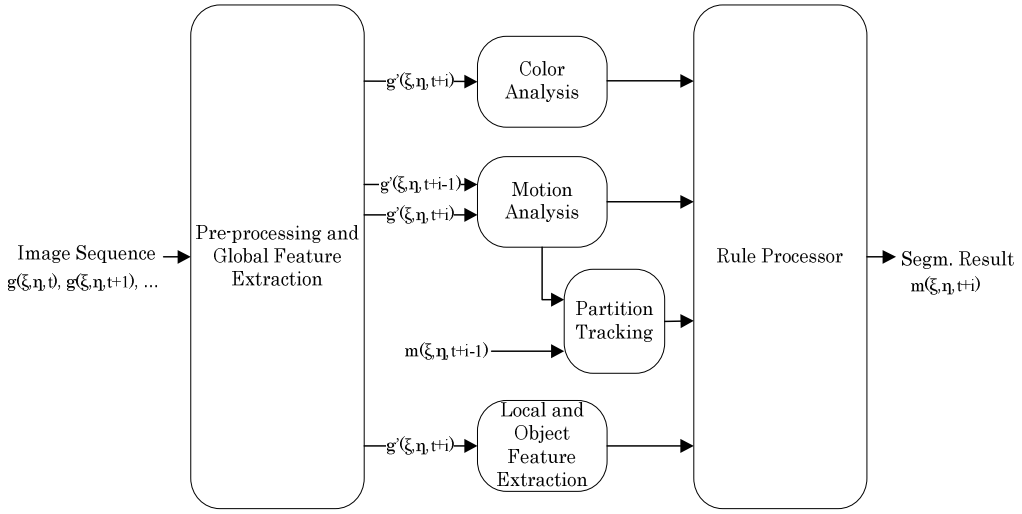


Fig. 16 – The simplified European COST 211quat Analysis Model

Note that the AM framework is more complex than depicted in Fig. 16, as features such as user interaction are supported. Only the parts of the AM relating to fully automatic video segmentation will yet be considered in this work. The corresponding modules will be presented in the following.

4.2.3.1 Global Motion Estimation and Compensation

The apparent camera motion is estimated and compensated given successive picture pairs. The estimation of the former is based on regression and is done under usage of the identified background samples [96],[98]. In case the background samples are unknown, as is the case in the first picture of the sequence or at scene cuts, samples at the picture borders are used as observation points. The assumption being that no moving object is located within a predefined margin in the concerned pictures. Motion is described using the perspective model (30). No motion compensation is done in case of inexistent camera motion. The latter is assumed if the estimated motion parameters $(a_1,\ldots,a_8)$ are of a magnitude lower than a given threshold. In case of existent camera motion, compensation is done with the estimated motion parameters and model failures of background samples are identified [106]. A posteriori displacement vector correction is operated at afore said locations by doing a full search within a restricted search area, which can only alleviate small model failures.

4.2.3.2 Scene Cut Detection

Scene cut detection, also called temporal segmentation in the literature, involves the identification of abrupt or gradual transitions in a video sequence. The AM uses the algorithm described in [96],[98], which basically analyzes the MSE of the luminance values of motion compensated background samples. Motion compensation of background samples is done using the segment tracking module described below. The formal definition of the scene cut criterion can be given as

$$\mathrm{E}_{\mathrm{cd}} = \frac{1}{N_{cd}} \sum_{\{(\xi,\eta)|\mathrm{cl}(\,\mathrm{m}(\xi,\eta,t+p\text{-}1)\,)=0\}} \left[\mathrm{I}_{\mathrm{cd}}(\xi,\eta,t+p) - \mathrm{I}_{\mathrm{cd}}^{\mathrm{gmc}}(\xi,\eta,t+p)\right]^2 > T_{cd} \text{ with } p = 0,\ldots,N_f^{cd} \quad (48)$$

with N_{cd} representing the number of background samples in the original picture I_{cd} and the motion compensated picture $\mathrm{I}_{\mathrm{cd}}^{\mathrm{gmc}}$ at time instance $t+p\text{-}1$. The current image is thereby given at time instance $t+p$. The background samples are labeled with zero in the AM framework. N_f^{cd} represents the number of pictures in the video sequence, while T_{cd} is the threshold that indicates a scene cut if the costs E_{cd} are higher.

4.2.3.3 Change Detection

The change detection algorithm used in the AM can be divided into four major steps. The first one corresponds to the initialization of the Change Detection Mask (CDM), while the second step consists in the relaxation of the initial CDM for spatial homogeneity. The third step focuses on the detection of moving shadow regions, while the final step deals with the temporal coherency of the object shapes.

The initial change detection mask is determined based on the approach proposed in [101]. The motion compensated, squared difference between successive pictures is initially determined. The "changed" areas are detected by applying a threshold operation on the difference pictures. Camera noise is thereby assumed to be Gaussian with a variance half as large as the squared difference's variance. The threshold used is fixed for a given picture resolution, e.g. 220 for QCIF and 165 for CIF sequences.

In the second step, spatial homogeneity of the CDM is ensured based on an approach presented in [101],[138]. This is done by iteratively reassigning the boundary samples of the identified moving regions to either the "changed" or "unchanged" class. For each border sample, a local threshold is determined based on the corresponding neighborhood. The reclassification of the boundary samples relies on a maximum a posteriori or MAP detector [106].

Due to the fact that spurious moving areas can be entailed by moving shadows, the AM operates a shadow detection at this stage for post-correction of the CDM. A detailed description of the shadow detection algorithm can be found in [106]. Note that the shadow subtraction step is optional in this framework and can be by-passed if needed.

The final step of the change detection operation consists in ensuring the temporal consistency of the identified object shapes. For that, a sample is labeled "changed" in the current CDM, if it had that same label in at least one of the past T_{cd}^{past} pictures. It is recommended to set T_{cd}^{past} to a large value in case of an input video sequence with low motion activity and vice versa [106]. In this framework, T_{cd}^{past} is set adaptively throughout the sequence based on a heuristic definition of T_{cd}^{past} as a function of the motion activity and the size of the objects [106],[131].

4.2.3.4 Color Segmentation

The color segmentation algorithm consists of two major sub-modules. The first one being a full-search block matching module used for segment tracking [106]. The latter is done through motion compensation of the color segmentation masks. Deformations of segment contours are considered by approximating them using $4x4$ blocks, which are motion compensated.

The second sub-module is the color segmentation algorithm. It is initialized with the motion compensated mask of the previous picture if both pictures belong to the same scene. In the case of the first picture (after a scene cut), motion compensation is skipped. The segmentation further proceeds with a color-based partition strategy, i.e. the luminance and chrominance difference between neighboring samples is submitted to a threshold. This step is followed by a split and merge segmentation approach [106]. Uncovered areas are assumed to yield

high prediction errors in this framework. The labeling of these areas is done using a watershed algorithm [106], which is basically a region growing approach. Finally, the region boundaries are fine-tuned.

As already said before, the segment tracking module is also used for scene cut detection. Note that the motion vectors determined by the segment tracking module are used as input for the local motion analysis module described below.

4.2.3.5 Local Motion Analysis

The local motion analysis module estimates the dense motion vector field of the input sequence. For that, a hierarchical block matching algorithm is used. The estimated block motion vectors are interpolated to obtain a dense motion field [106],[131].

4.2.3.6 Rule Processor

The rule processor combines the results obtained through change detection, color segmentation and local motion analysis to achieve the final binary mask with the refined moving and background textures. The merger of the multi-feature analysis results is basically done in a two step procedure.

The first step consists in eliminating the uncovered areas in the change detection mask. That is change detection and motion analysis information is first combined. A foreground or "changed" sample remains "changed" if and only if the starting and end point of the corresponding motion vector lie in the "changed" area of the change detection mask. The sample is switched to "unchanged", i.e. background, otherwise.

The second step tackles boundary accuracy issues. Here, color segmentation boundaries are assumed to be very accurate and are thus transferred to the corrected change detection mask. As a consequence thereof, a reclassification of some samples may be required [106].

4.2.4 Proposed Spatio-Temporal Segmentation Algorithm

4.2.4.1 Principle

A spatio-temporal, parametric segmentation algorithm has been developed in this thesis. It is inspired by the work of Adiv [119] that has influenced the formulation of various video segmentation algorithms [120],[15],[121],[122],[123]. The principle of the algorithm developed in this thesis is depicted in Fig. 17. As can be seen, the proposed approach can be resumed as a split and merge segmentation strategy with tracking abilities. That is, at a given picture transition, the optical flow field is split into homogeneously moving regions using robust statistics, namely a maximum-likelihood estimator called M-estimator. The optical flow and subsequent M-estimation can be initialized with segmentation masks delivered by a spatial segmentation module. M-estimation is then done for each of the spatial regions separately. The initialization of the motion splitter module through spatial segmentation relies on the fact that good initial segments typically yield good motion estimation. Note that on the other hand, a good motion estimation yields successful segmentation, a classical chicken and egg dilemma.

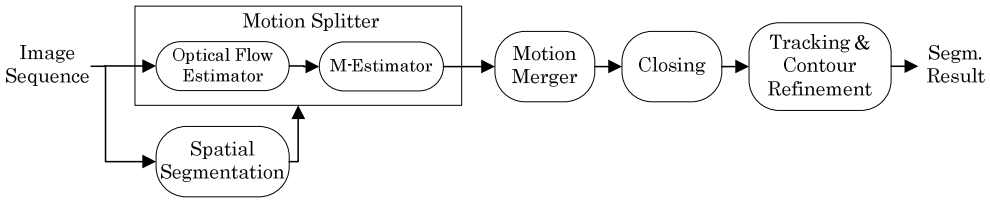


Fig. 17 – Block diagram of the proposed spatio-temporal segmentation algorithm

The typically over-segmented masks obtained after the splitting step are further processed by the motion merger module, which aims to convey all regions featuring similar motion properties to the same class. Note that this is consistent with the content-based video coding idea proposed in this thesis (cp. Sec. 3.3).

After the merging step, a morphological closing operation [4] is applied to remove small “holes”, i.e. small clusters with given labels, located within a much larger homogeneous texture with a different label. Finally, temporal tracking as well as contour refinement of the detected regions are performed. The output of the

proposed segmentation algorithm is a mask sequence showing the location of homogeneous spatio-temporal segments in the input video sequence.

The proposed spatio-temporal segmentation algorithm will be called the algorithm in the following in the event of unambiguous context. Note that the spatial segmentation operation depicted in Fig. 17 corresponds to the algorithms presented in Sec. 4.1. For that, the former will not be considered in the following. The optical flow estimator corresponds to the algorithm by Black and Anandan [41] that was introduced in Sec. 4.2.1. Hence, it will not be further detailed in the following either.

4.2.4.2 M-Estimation

Maximum likelihood approaches are the most commonly used methods for model fitting tasks. The likelihood of the model given the data is maximized in this context. A robust M-estimator eliminates potential outliers a posteriori, i.e. without a priori knowledge of the data.

In the context of the spatio-temporal texture analyzer, the data to analyze correspond to a motion field determined by the optical flow estimator at a given image transition. The perspective motion model (30) is fitted to the motion field in this context. Outlier motion vectors are identified and their bias on the outcome reduced by the M-estimator in the course of the global motion estimation process. This is done by assigning a weight, inversely proportional to the fitting accuracy, to each motion vector. The weights are real numbers normalized to the interval $[0,1]$ and exactly zero only in case of very crude outliers.

The formalization of the M-estimator used in this thesis and described in [78] will be presented in the following. The motion field is first modeled using the estimated motion parameters. Given the perspective motion model (30), the following iterative algorithm can be formulated

$$\boldsymbol{\omega}^{(n)}(\tau)=\begin{pmatrix}\omega_x^{(n)}(\tau)\\ \omega_y^{(n)}(\tau)\end{pmatrix}=\mathrm{h}\big(\mathbf{p}(\tau)\big)=\begin{pmatrix}\dfrac{a_1(\tau)+a_3(\tau)x^{(n)}+a_4(\tau)y^{(n)}}{1+a_7(\tau)x^{(n)}+a_8(\tau)y^{(n)}}-x^{(n)}\\ \\ \dfrac{a_2(\tau)+a_5(\tau)x^{(n)}+a_6(\tau)y^{(n)}}{1+a_7(\tau)x^{(n)}+a_8(\tau)y^{(n)}}-y^{(n)}\end{pmatrix} \tag{49}$$

where $\left(\omega_x^{(n)}(\tau),\omega_y^{(n)}(\tau)\right)$ correspond to the predicted motion vectors at iteration step τ, while $\mathbf{p}(\tau)$ is the motion parameter set (31) and $\left(x^{(n)},y^{(n)}\right)$ are the considered sample locations. In a subsequent step, the deviations between estimated motion vectors $\left(\omega_x^{(n)}(\tau),\omega_y^{(n)}(\tau)\right)$ and reference motion vectors $\left(v_x^{(n)},v_y^{(n)}\right)$ (optical flow estimation) are determined at each pixel location as

$$e^{(n)}(\tau)=\left|v_x^{(n)}-\omega_x^{(n)}(\tau)\right|+\left|v_y^{(n)}-\omega_y^{(n)}(\tau)\right| \ . \tag{50}$$

The mean error thereby corresponds to

$$\mu_e(\tau)=\frac{1}{N_{me}}\sum_{n=1}^{N_{me}}e^{(n)}(\tau) \tag{51}$$

where N_{me} represents the number of samples. The weights for each motion vector can now be determined as

Error! Objects cannot be created from editing field codes. . (52)

where q_{me} is a degree of freedom of the M-estimator that steers the outlier threshold. The larger q_{me}, the more does the robust approach behave as (37), i.e. the more sensitive does it become to outliers. On the other hand, the smaller q_{me}, the more conservative does the system become with regard to outliers. In case q_{me} is too small, no or not enough inliers are found to determine the motion parameters. Following the recommendation in [78], q_{me} is set to three in this thesis. A robust reformulation of (37) can now be given as

$$\mathbf{p}(\tau)=\left(\mathbf{D}^{\mathrm{T}}\,\mathbf{W}(\tau)\,\mathbf{D}\right)^{-1}\mathbf{D}^{\mathrm{T}}\,\mathbf{W}(\tau)\mathbf{k} \ . \tag{53}$$

Outliers are assigned low weights $\mathbf{W}(\tau)$ by this approach. Hence, their influence on the global motion estimation process is reduced. The weight matrix can be written as

$$\mathbf{W}(\tau)=\begin{pmatrix} w^{(1)}(\tau) & 0 & \cdots & \cdots & 0 \\ 0 & w^{(1)}(\tau) & & & \vdots \\ \vdots & & \ddots & & \vdots \\ \vdots & & & w^{(n)}(\tau) & 0 \\ 0 & \cdots & \cdots & 0 & w^{(n)}(\tau) \end{pmatrix}. \tag{54}$$

Each weight appears twice in the weight matrix as, in matrix $\mathbf{D}$, each data influences two rows of the matrix.

As can be seen above, the motion estimation approach via M-estimation is an iterative method. The estimation is stopped either if the mean prediction error (51) is smaller than a given threshold or if a maximum number of iterations is reached.

4.2.4.3 Motion Splitter Principle

The motion splitting module initially operates dense optical flow estimation at image transitions using the algorithm by Black and Anandan [41] presented in Sec. 4.2.1.

The M-estimator is then applied to the dense motion field in order to identify homogeneously moving regions. Homogeneity is thereby defined with regard to the perspective motion model (30) as described above. Initialization of the M-estimator can be done by the spatial texture analysis module. In this case, M-estimation is done for each region identified by that analyzer (cp. Fig. 17). In case no spatial analysis is conducted, M-estimation is done based on the entire motion field. Although a robust M-estimator is used, hints provided by the spatial texture analyzer typically yield an improved segmentation of the motion field.

M-estimation is executed recursively by the motion splitter module. The given motion field is first split into inliers and outliers. Typically, outliers indicate either multiple motions in the considered region or optical flow estimation failures. In the latter case, scattered outliers can usually be observed, while the former case typically reveals larger connected outlier areas. An example of optical flow estimation failure can be seen in Fig. 18 at the picture borders. M-estimation is applied to the outliers if and only if they represent a significant

area, which is assumed to be verified if their amount exceeds a given threshold. This can be formalized as follows

$$\left|O_p\right| \geq T_{ms} \tag{55}$$

where O_p corresponds to the current (p^{th}) outlier set, T_{ms} is the outlier threshold, and $\left|\ \right|$ is the size of O_p. This process is applied recursively until (55) is violated. Outlier regions that are smaller than T_{ms} are attributed a reserved label, e.g. zero. Note that for each inlier region, the corresponding sample coordinates, the perspective motion parameters (30) as well as the mean error (51) are retained.

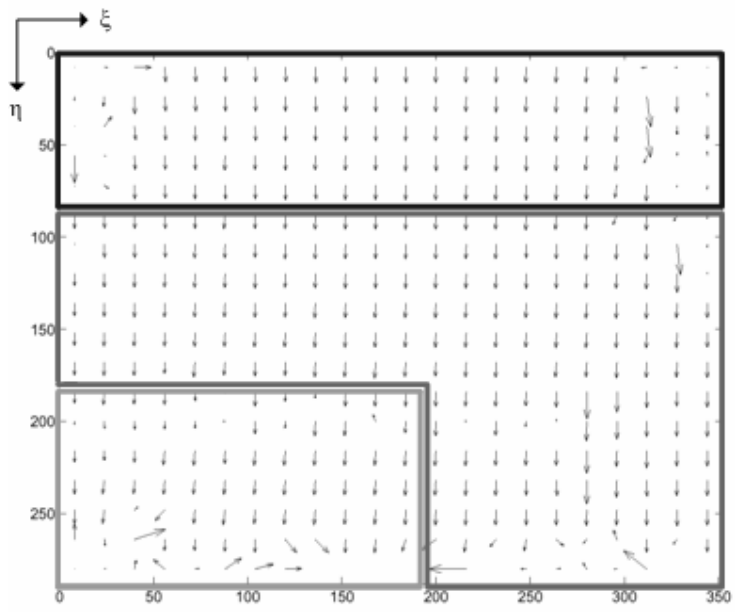


Fig. 18 – Principle of the motion splitter module

Fig. 18 depicts the functionality of the motion splitter module. It is assumed that initialization is provided by the spatial texture analysis module. The three spatial regions identified by the corresponding analysis module are shown as black and gray frames in Fig. 18. As explained above, each of the spatial regions is individually analyzed by the motion splitter, which can result in splitting each of the initial spatial regions into several, smaller, spatio-temporally homogeneous textures.

4.2.4.4 Motion Merger

Applying the motion splitter module on the whole motion field without initialization by the spatial texture analysis module ideally yields regions with pairwise distinct motion properties. In the event of preliminary spatial region initialization, motion splitting can yield distinct regions with similar motion properties. In order to account for this over-segmentation eventuality, motion merging is required.

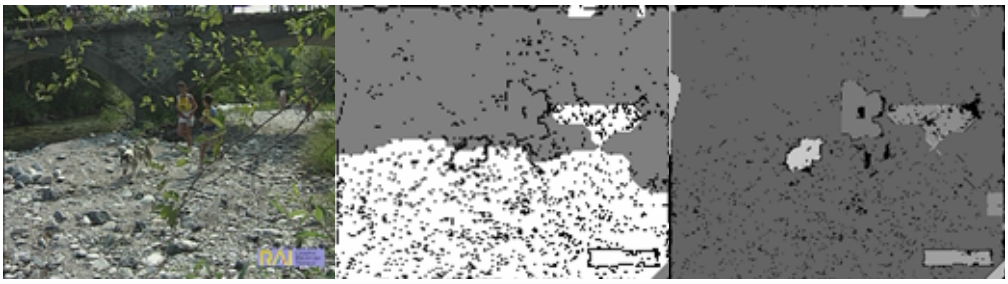


Fig. 19 – Segmentation masks of the "Husky" test sequence (leftmost image) before (mid mask) and after (rightmost mask) the merging operation

The motion merger module fuses regions with similar motion properties. Initially, the homogeneous regions are sorted in descending order with regard to their size. The merger of a pair of regions with respective mean modeling errors μ_e^1 and μ_e^2 (51) is then simulated. The mean modeling error $\mu_e^{1,2}$ of the merged region is determined. Large regions are first compared to and eventually merged with smaller once. Two regions are hereby assumed to feature similar motion properties if $\mu_e^{1,2}$ is not larger than the mean errors μ_e^1 and μ_e^2 of the single regions. In case the modeling costs increase due to the merger, the considered regions are not fused. Merged regions are assigned the modeling error $\mu_e^{1,2}$ otherwise. The merge criterion can be formalized as follows

$$\text{If } \mu_e^{1,2} \leq \mu_e^1 \wedge \mu_e^{1,2} \leq \mu_e^2 \Rightarrow R = R_1 \cup R_2 \tag{56}$$

where R corresponds to the merged region and R_1 and R_2 are the single regions. Fig. 19 depicts segmentation masks obtained for the same image of the Husky test sequence. The mid mask corresponds to the output of the motion splitter module. As can be seen, two large regions are identified within the

homogeneously moving background, which clearly corresponds to over-segmentation. The right mask shows the result obtained by applying the merging algorithm on the left mask. It can be seen that the two background regions have been fused. Nevertheless are still a number of "holes" visible that correspond to noise.

4.2.4.5 Morphological Closing Operation

After the motion splitting and merging steps, the segmentation masks typically exhibit a number of "holes". That is, scattered small clusters within much larger ones can be observed. The latter typically relate to optical flow estimation errors or modeling inaccuracies in the M-estimation process. They can also be due to over-sensitivity (cp. q_{me} (52)) or very small regions that could not be analyzed by the M-estimator.

The achievement of larger homogeneous areas is enforced by applying a closing operation on the output of the merger module. The "holes" are thereby marked with a reserved label, e.g. label zero. The morphological closing operation is then conducted at those locations and within the limits of larger homogeneous areas to avoid shifting major motion borders in the masks. The advantage of the closing operation in this context is that larger "holes", assumingly having a high likelihood to indicate real local motion activity, are ideally kept as is, while smaller ones are closed, i.e. assigned the same label as the surrounding texture.

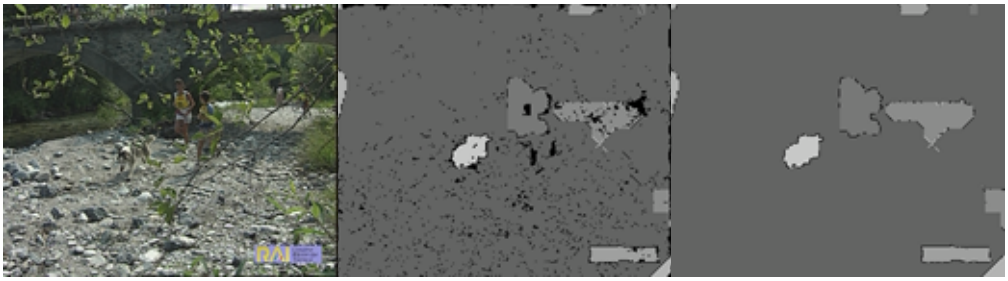


Fig. 20 – Segmentation masks of the "Husky" test sequence (leftmost image) before (mid mask) and after (rightmost mask) the morphological closing operation

Fig. 20 depicts segmentation masks obtained for the same image of the Husky test sequence. The mid mask corresponds to the output of the merger module. As can be seen, a large number of "holes" exist within the homogeneously moving

background. The right mask shows the result obtained by applying the closing algorithm on the left mask. It can be seen that the "holes" have been eliminated.

4.2.4.6 Tracking and Contour Refinement

Up to this stage, the segmentation masks have been generated at image transitions. That is, the label assignments are only consistent for contiguous image pairs. In order to extend label consistency to the entire input sequence, a tracking module is required. For that, the textures identified in the course of the video sequence are indexed. Each texture found in the sequence is matched with the indexed textures. In case it is already known, the corresponding label is assigned to it, the texture is indexed as new otherwise and assigned a new label.

The Scalable Color descriptor [2] is used for similarity estimation. It is basically a color histogram in the HSV color space. Note that the SCC descriptor can in principle be used in combination with other color spaces. Two textures are considered to be similar if the distance between their feature vectors lies below a given threshold:

$$\mathrm{d}_{\ell_1}\left(\mathbf{f}_{scc}^{1}, \mathbf{f}_{scc}^{2}\right) \leq T_{scc} \tag{57}$$

where $\mathrm{d}_{\ell_1}()$ represents the ℓ_1 metric, $\mathbf{f}_{scc}^{p}$ $(p = 1,2)$ are the feature vectors of the considered textures, while T_{scc} is the similarity threshold. Notice that other similarity measures, as the Earth Mover's Distance (EMD) [139], can be used instead of ℓ_1 in order to enable some invariance against luminance and saturation variations that are entailed by effects as shadowing or reflections for instance. The functionality of the tracking module is depicted in Fig. 21.

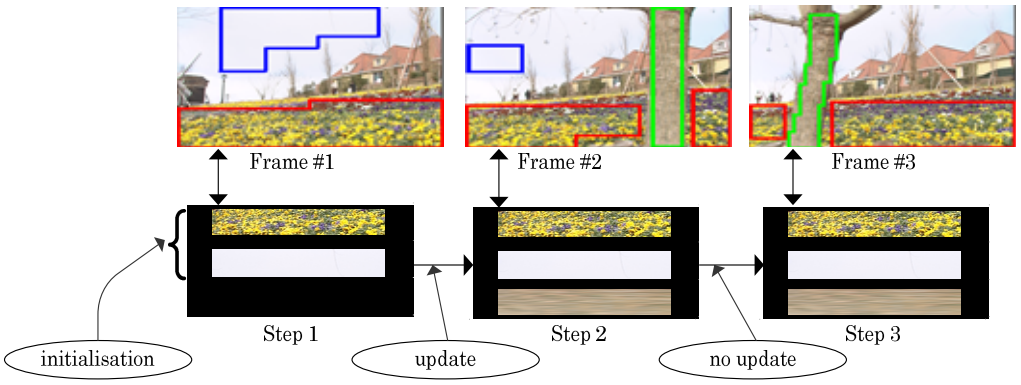


Fig. 21 – Principle of the tracking module

As already said above, the motion analysis conducted up to this stage has been done for image pairs. Hence, no use is made of the prior knowledge related to the segmentation of previous image pairs. The contour refinement module tackles this issue by applying a temporal median filtering (TF) algorithm on the output of the tracking module. The usage of this approach for contour refinement is motivated by the fact that it enhances noisy images and preserves edge information [16]. Temporal median filtering is operated on the segmentation masks. A freely settable amount of motion compensated masks preceding the current mask is thereby considered. This operation yields a label update in the current picture with the label of the majority of the past pictures at the specified location. This contributes to stabilize object shapes in the course of the video sequence. On the other hand, at scene changes or in case of fast motion, wrong masks may be generated. For that, in this work, the costs of the masks generated through temporal filtering are compared to those of the masks without temporal filtering. The masks with the lowest costs in terms of modeling inaccuracies are kept.

$$\begin{aligned} \mu_e^{tm} < \mu_e^{no\_tm} &\Rightarrow R_{tm} \\ \mu_e^{no\_tm} \leq \mu_e^{tm} &\Rightarrow R_{no\_tm} \end{aligned} \quad . \tag{58}$$

The decision criterion between temporal mapping and no temporal mapping is formalized in (58). μ_e^{tm} corresponds to the mean modeling error for a single picture transition and temporal mapping. $\mu_e^{no\_tm}$ is the mean error for a single

picture transition and no temporal mapping. The regions obtained before and after temporal mapping are referred to as R_{no_tm} and R_{tm} respectively.

4.3 Objective Evaluation of Segmentation Masks

4.3.1 Huang and Dom's Measures

Following the work by Freixenet et al. [50], segmentation results shall be evaluated based on quantitative methods proposed by Huang and Dom [63] in this thesis. The parameters for performance evaluation against ground truth are used, i.e. it is assumed that the true segmentation masks of given ground truth images are known.

Huang and Dom [63] split the automatic segmentation evaluation into two parts. The first one consists in a boundary-based evaluation, while the second one corresponds to a region-based quality assessment approach. Both parameter classes will be presented into detail in the following.

4.3.1.1 Weighted Boundary Error Rates

Boundary error rates measure the discrepancy between true and segmented region contours. It is assumed that G_b is the set of true boundaries, while the set of segmented contours is called S_b. Let S_b^{tp} stand for the set of true positive contour pixels and S_b^{fp} for the set of false positive samples. S_b^{tp} and S_b^{fp} can then be formally defined as

$$S_b^{tp} = \left\{ (\xi,\eta) \,\middle|\, (\xi,\eta) \in G_b \wedge (\xi,\eta) \in S_b \right\} \tag{59}$$

and

$$S_b^{fp} = \left\{ (\xi,\eta) \,\middle|\, (\xi,\eta) \notin G_b \wedge (\xi,\eta) \in S_b \right\} \tag{60}$$

respectively, where (ξ,η) correspond to the coordinates of a given sample. The missing (e_b^m) and the false boundary (e_b^f) rates can be derived from above definitions as follows

$$e_b^m = \frac{\left|G_b\right| - \left|S_b^{tp}\right|}{\left|G_b\right|}, \tag{61}$$

$$e_b^f = \frac{\left| S_b^{fp} \right|}{\left| G_b \right|} \tag{62}$$

where $\left| \; \right|$ denotes the size of a given set. The false boundary rate can also be seen as a false positives rate r_b^{fp}, while the true positives rate r_b^{tp} can be derived from the missing boundary rate as follows

$$r_b^{tp} = 1 - e_b^m = \frac{\left| S_b^{tp} \right|}{\left| G_b \right|} \; . \tag{63}$$

The objective segmentation quality measures defined in (61), (62), and (63) capture the proportion of misclassified image samples. Nonetheless, they can not quantify the "closeness" of the location of the true and the segmented region contours. Hence, Huang and Dom [63] introduced weights w_b^m and w_b^f to the missing and false boundary rates respectively, where the weights represent the mean distance between the misclassified samples and the ground truth boundary. The distance between a point and a set is thereby defined as the minimum absolute distance from the point to all points of the considered set. To fully describe the boundary accuracy of a given segmentation result the pairs (e_b^m, w_b^m) and (e_b^f, w_b^f) are needed. Note that w_b^m and w_b^f are normalized with the length of the main diagonal of the considered image in this work to achieve a range from 0.0 to 1.0 for the weights.

4.3.1.2 Region Error Rates

Region error rates measure the discrepancy between true and segmented regions. This is done by applying the directional Hamming distance [63] $d_H(S_1 \rightarrow S_2)$ from a segmentation $S_1 = \{R_{11}, R_{12}, R_{13}, \ldots\}$ to another segmentation $S_2 = \{R_{21}, R_{22}, R_{23}, \ldots\}$. For that, mapping of the regions of S_2 onto those of S_1 is required and realized such that the region R_{2n} is associated with the region R_{1p} if and only if $R_{2n} \cap R_{1p}$ is maximal. The directional Hamming distance then measures the total area for which $R_{2n} \cap R_{1p'}$ is non-maximal. The formal definition of the directional Hamming distance can be expressed as follows

$$d_H(S_1 \rightarrow S_2) = \sum_{R_{2n} \in S_2} \sum_{R_{1p'} \in S_1, R_{1p'} \neq R_{1p}, R_{1p'} \cap R_{2n} \neq \emptyset} \left| R_{2n} \cap R_{1p'} \right|. \quad (64)$$

$$e_r^m = \frac{d_H(S_r \rightarrow G_r)}{\left| S_r \right|} \quad (65)$$

$$e_r^f = \frac{d_H(G_r \rightarrow S_r)}{\left| S_r \right|} \quad (66)$$

Given the definition of the Hamming distance, Huang and Dom [63] define a missing and a false alarm rate denoted e_r^m and e_r^f respectively. It is assumed that G_r is the set of true regions, while the set of segmented regions is called S_r. e_r^m then measures the percentage of the samples in G_r being mistakenly segmented into wrong regions in S_r, while e_r^f describes the percentage of image samples in S_r falling into regions of G_r that are non-maximal intersected with the region under consideration. The formalization of the missing and the false alarm error rates is given in (65) and (66). Note that $\left| S_r \right|$ corresponds to the considered image size.

$$r_r^{tp} = 1 - e_r^m \quad (67)$$

Similarly to the boundary error rates, the false alarm rate can be seen as a false positives rate r_r^{fp}, while the true positives rate r_r^{tp} can be derived from the missing rate as shown in (67).

4.3.2 Receiver Operating Characteristic Curve

Receiver Operating Characteristic (ROC) curves were developed in the 1950's in the field of radio signal processing [140]. In this work, ROC curves are used in the context of image segmentation to evaluate the accuracy of a given segmentation approach given a ground truth set, i.e. the ability of the segmentation algorithm to provide accurate image partitions. The abscissa of the ROC curve represents the false positive (FP) rate, while the ordinate corresponds to the true positive (TP) rate. (FP,TP) pairs are measured for several configurations of the segmentation algorithm, each pair corresponding to the average performance of the considered approach over the ground truth set. The area under the ROC curve, also called AUC, can be interpreted as the percentage

of randomly drawn data pairs (one from the true positive and one from false positive class) for which the segmentation algorithm yields a correct classification. The accuracy of the segmentation approach is proportional to AUC. The ideal AUC corresponds to 1.0, i.e. 100% TP independently of the FP rate. An AUC of 50% stands for a worthless segmentation approach, while 70%-80% are obtained for a fair, 80%-90% for a good, and 90%-100% for a very good segmentation algorithm.

4.4 Experimental Results

4.4.1 Spatial Texture Analysis

The evaluation of the spatial texture analysis module is conducted with a ground truth set of 100 images from the Corel Gallery™ (US version, 07/1998) database. The images are selected in consideration of the lighting conditions, the presence/absence of details in the images and a "good" coverage of the HSV color space. A further discrimination criterion consists in selecting only images that show a large variance in more than one dimension in the HSV color space. This is an important prerequisite to ensure meaningful results with regard to the best optimization algorithm. For each ground truth image, a reference partition is generated manually. The clusters thereby reflect a semantic decomposition of the scene. Each ground truth image is also segmented using the automatic segmentation approaches presented in Sec. 4.1.2, 4.1.4, and 4.1.5. The obtained partitions are compared to the reference partitions using the measures defined in Sec. 4.3. Each of the automatic segmentation algorithms features two degrees of freedom, i.e. the window size m and the number of pyramid levels ℓ' (cp. Sec. 4.1.2).

Experiments show that ℓ' can be set to four without penalizing any of the ground truth images. Moreover, it was found that m has the most significant impact on segmentation quality compared to ℓ'. For that, m is drawn from the set $\{1,5,10,15,20,25,30,35,40,45\}$ in the experiments. The results of the evaluations are depicted in Fig. 22 and Fig. 23 via ROC curves for boundary and cluster quality assessment respectively. The AUC values are given in both figures for each optimization algorithm.

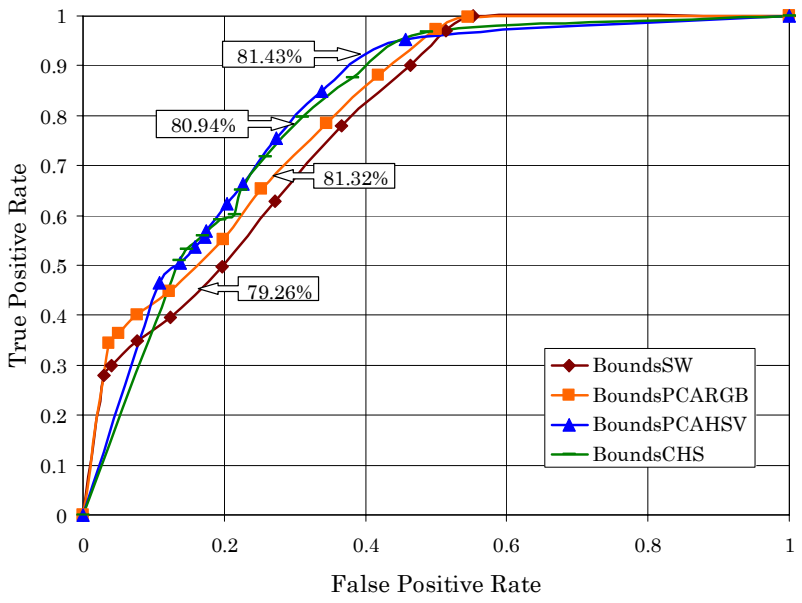


Fig. 22 – ROC curves for boundary error rates

It can be seen that better results than the genuine segmentation approach by Spann and Wilson [56] (SW) are achieved for each of the optimization algorithms. This shows that the algorithms proposed in this thesis are more powerful than [56] (SW) for color images. PCAHSV, which corresponds to redundancy elimination in the HSV color space (cp. Sec. 4.1.5), yields the best results both in terms of region (83.5% AUC) and boundary (81.43% AUC) accuracy. The other optimized methods, i.e. redundancy elimination in the RGB color space (PCARGB) and color channel pruning (CHS, cp. Sec. 4.1.4) give slightly worse results.

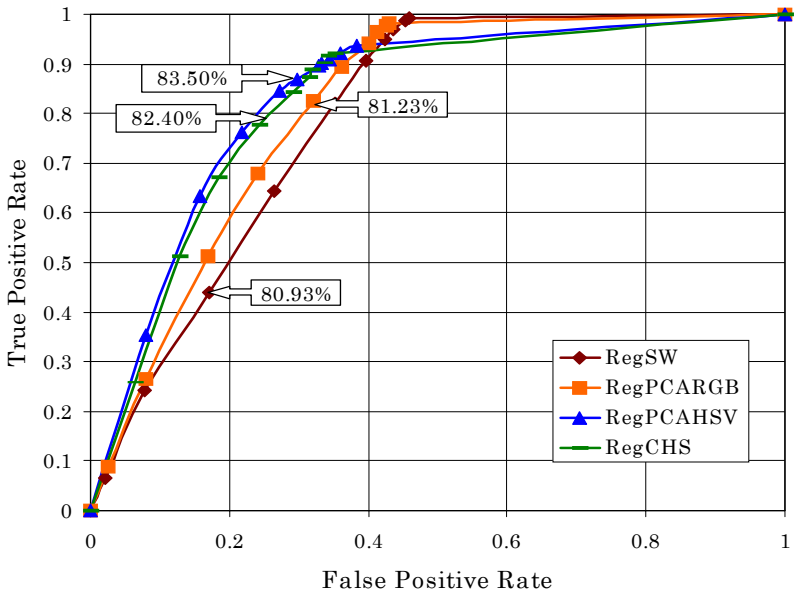


Fig. 23 – ROC curves for region error rates

Boundary accuracy is improved from fair to good, compared to [56], in all cases (cp. Sec. 4.3.2). Fig. 24 and Fig. 25 depict the evaluation results w.r.t. w_b^m and w_b^f. Each box plot corresponds to 1000 data samples obtained for 100 test images and 10 settings of parameter m. The box plot representation is well suited for this analysis, as it allows for good visualization of mean and variance of the data samples. The red, horizontal line within a box represents the median of the corresponding data samples.

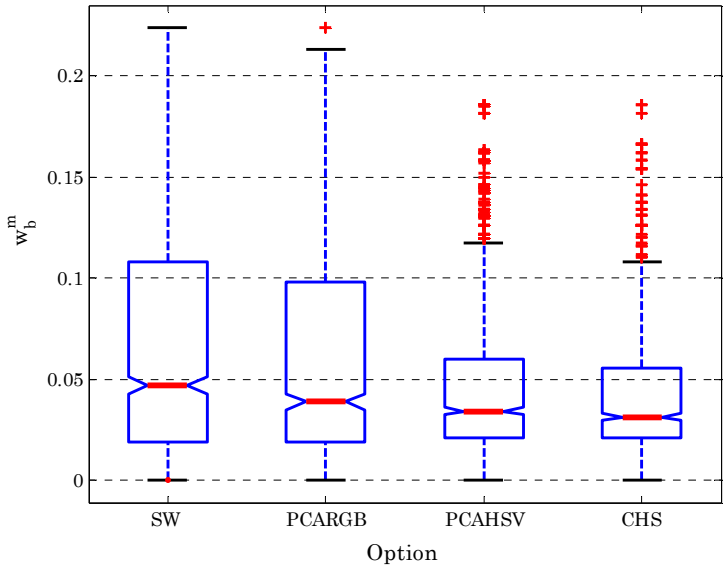


Fig. 24 – Box plots of missing boundary error rate weights for all options

Each box is drawn from the lower ($\upsilon_{0.25}$) to the upper ($\upsilon_{0.75}$) quartile. The so-called whiskers are drawn from the lower (upper) quartile to the smallest (largest) measured weight and thus cover the full span of the given data. Weights smaller than $\upsilon_{0.25} - 1.5\Delta_{QR}$ or greater than $\upsilon_{0.75} + 1.5\Delta_{QR}$, with

$$\Delta_{QR} = \upsilon_{0.75} - \upsilon_{0.25} \,, \tag{68}$$

are called outliers.

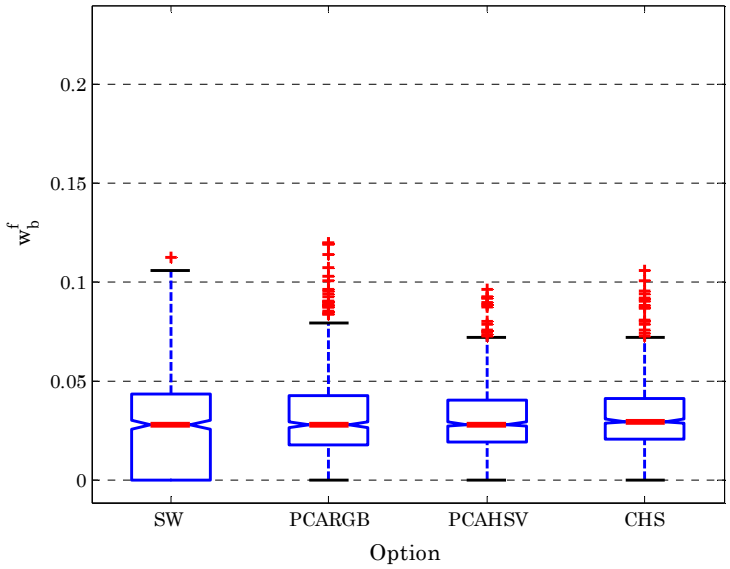


Fig. 25 – Box plots of false boundary error rate weights for all options

Δ_{QR} represents a measure for the standard deviation of the middle 50% of the data. Outliers are depicted as „+“ in Fig. 24 and Fig. 25. It can be seen in Fig. 24 that the mean distance between the missing and the automatically determined boundary samples (w_b^m) is smaller than the anchor's for all optimization approaches.

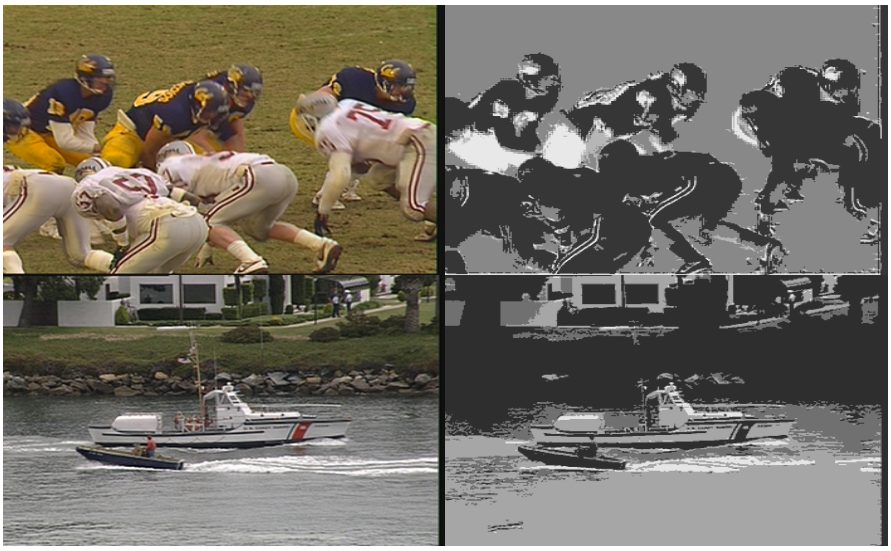


Fig. 26 – Exemplary segmentation results obtained with the channel pruning (top row) and the redundancy elimination algorithms ($m = 35$, $\ell' = 4$)

The variance of w_b^m is globally smaller than the anchor's. The measured gains can be seen as statistically significant where there is no overlap of the notches (cp. CHS and SW in Fig. 24), which is basically the case for all optimization strategies w.r.t. SW. The mean w_b^f is almost the same for all options as can be seen in Fig. 25.

The best overall results are obtained for PCAHSV and CHS. CHS is seen to be better than PCARGB because although the boundary AUC of the former is smaller than the latter's, the closeness of the CHS boundary samples to the reference boundary is significantly higher compared to PCARGB and SW. Some segmentation results are depicted in Fig. 26 for key pictures of the "Football" and "Coast guard" test sequences.

4.4.2 Spatio-Temporal Texture Analysis

The evaluation of the spatio-temporal texture analysis algorithm presented in this work is conducted by operating a comparative evaluation w.r.t. the COST 211quat Analysis Model (AM, cp. Sec. 4.2.3). For consistency reasons (cp. Sec. 4.4.1), the quality measures presented in Sec. 4.3.1 will be used for evaluation of the results obtained with the spatio-temporal segmentation algorithm. The author is aware of the fact that other measures have been proposed within the COST 211quat framework. However, they will not be considered as they are not expected to provide decisively more information than the measures used in this thesis. In fact, the COST 211quat spatial quality measures are similar to w_b^m and w_b^f.

Seven test sequences, at CIF resolution, are considered for the comparative evaluation. Namely "Bus", "Canoe", "Coast Guard", "Container Ship", "Football", "Foreman", and "Stefan". For "Coast Guard", "Foreman", and "Stefan", the segmentation masks provided by MPEG-4 are used as reference segmentation. For these video sequences, the discrepancy between the video object concept of MPEG-4 and the coding approach presented in this work is ignored for the sake of practicability (cp. Sec. 2.1 and Sec. 3.1). Manual segmentation, conforming to the coding approach presented in this thesis, is done for the remaining test sequences.

Parameter	Value	Semantics
mv_regions	4	Nr. of regions in the motion seg. mask
mot_res	4	Spatial resolution of motion vector field (mot_res x mot_res block)
mergesize	20	Merge regions smaller than this threshold with a neighboring region
gmec_iter	70	Nr. of iterations for global motion estimation (GME)
pel_count	10000	Number of observation points for GME
grad_x_min	0.0	Min. gradient in x dir. of observation points in GME
grad_x_max	260.0	Max. gradient in x dir. of observation points in GME
grad_y_min	0.0	Min. gradient in y dir. of observation points in GME
grad_y_max	260.0	Max. gradient in y dir. of observation points in GME
cdm_cif/ cdm_cif_mb	50.0	Change detection threshold
scd_thresh	250.0	Scene-cut detection threshold
afs	0	Skip *frameskip* pictures for motion estimation to enhance tracking of slow motion. Interpolate masks between key pictures.
frameskip	3	Nr. of skipped pictures
half_pix	1	Sub-pixel accuracy for motion vector estimation

Tab. 1 – Settings of the COST211quat AM

As, for both the COST AM and the algorithm presented in this thesis, the reduction of the influential degrees of freedom to one for each module can not reasonably be done, the ROC-related evaluation presented in Sec. 4.3.2 will not be carried out here. Alternatively, each of the texture analysis modules is tuned to achieve the best possible result for all test sequences given a single configuration. The respective configurations of the texture analysis modules are detailed in Tab. 1 and Tab. 2. Note that only the most important parameters can be found in the tables. The algorithm presented in this thesis will be referred to as the algorithm in the following.

Parameter	Value	Semantics
ℓ'	4	Nr. of pyramid levels (Spatial texture analysis)
m	35	Half window width (Spatial texture analysis)
num_closing_iters	20	Nr. of iterations of the morph. closing operation
sim_norm_scc	EMD	Similarity measure (Tracking module)
sim_thr_scc	0.12	Similarity threshold (Tracking module)
min_outlier_pels	256	Min. nr. of outlier pixels for M-estimation (Motion Splitter module)
q_{me}	3	Outlier threshold (M-estimator)
λ_D	10.0	Regularization parameter (Optical Flow Estimator)
θ_D	10.0	Data constraint's scale parameter (Optical Flow Estimator)
θ_S	1.0	Spatial constraint's scale parameter (Optical Flow Estimator)
black_res	4	Nr. of pyramid levels (Optical Flow Estimator)

Tab. 2 – Settings of the proposed spatio-temporal algorithm

For each of the considered test sequences, it is found that the corresponding e_b^m value, i.e. the missing boundary error rate, of the algorithm is significantly lower than the same error rate for the COST AM (cp. Fig. 27, Fig. 28, and Fig. 29). At the same time, the algorithm typically yields significantly lower w_b^m values compared to the AM. This shows that not only can ground truth boundaries be more accurately be found with the algorithm, but the missed boundary samples are also very much closer to the algorithm's boundaries compared to the AM (cp. Fig. 27, Fig. 28).

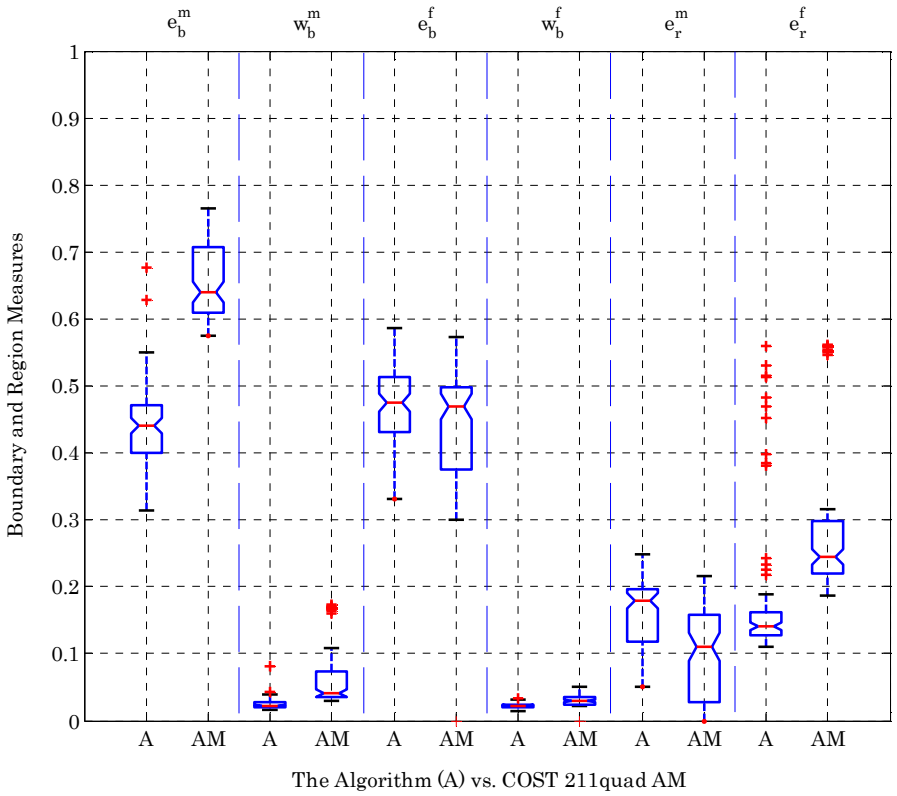


Fig. 27 – Boxplots of Huang and Dom's measures for the "Bus" sequence

However, the algorithm tends to generate more false boundary samples than the AM (cp. Fig. 27, Fig. 28), i.e. the algorithm yields a higher e_b^f value in most of the cases. This negative outcome is yet attenuated by the fact that the proximity (w_b^f) of the algorithm's false boundary samples to the ground truth boundaries is, in general, significantly higher compared to the AM. For the algorithm, the w_b^f variance is again always smaller than the corresponding AM variance. The boundary estimation evaluation can be summarized as follows: The algorithm finds more ground truth boundaries than the AM at the price, however, of additional erroneous boundary samples, which are typically relatively close to the ground truth boundaries.

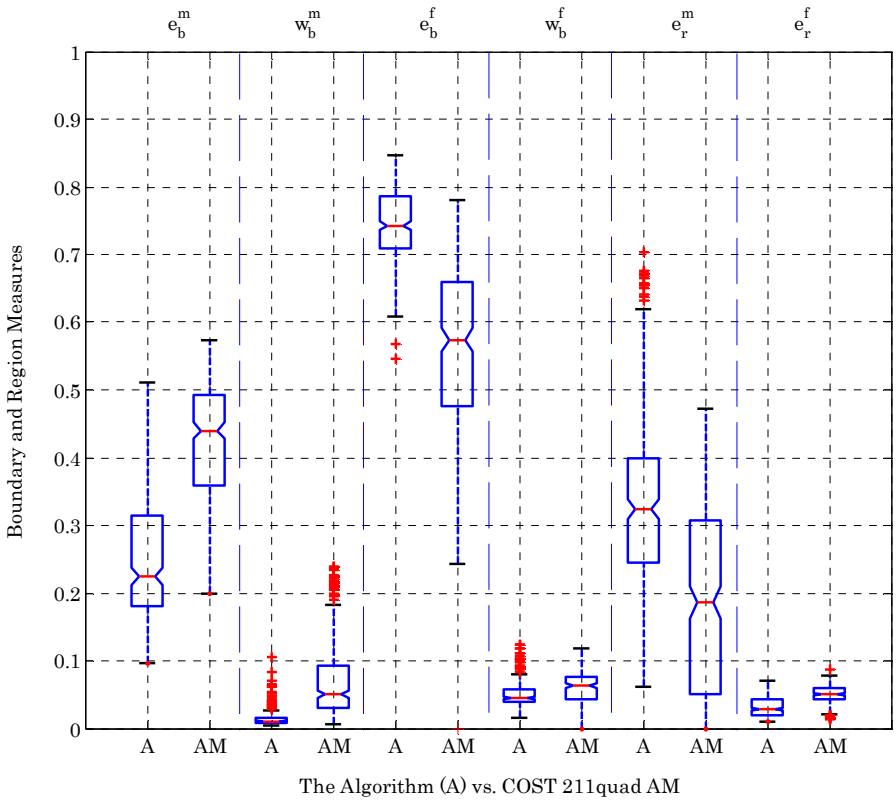


Fig. 28 – Boxplots of Huang and Dom's measures for the "Stefan" sequence

Region estimation evaluation shows that, in general, the false region error rate e_r^f is significantly lower for the algorithm than for the AM (cp. Fig. 27, Fig. 28, and Fig. 29). On the other hand, the missing region error rate e_r^m is typically significantly higher for the algorithm than for the AM. Given the definition of e_r^m and e_r^f, this indicates that the algorithm is more prone to over-segmentation than the AM. Notice that the results obtained for the "Canoe", the "Coast Guard", the "Football", and the "Foreman" sequences are depicted in Appendix A.

The summary of the boundary-based and region-based evaluations is that regions found by the algorithm are more reliable than those found by the AM, i.e. the probability that they can be found in the ground truth segmentation masks is higher for the algorithm. The true object boundaries are moreover better detected by the algorithm. However is the algorithm prone to over-segmentation. This can be explained by the merger procedure provided by the algorithm. That is, although smaller regions are in principle merged with larger ones, this is done

w.r.t. the ruling cost function defined in (51) as explained in Sec. 4.2.4. On the contrary to the algorithm, the AM performs an almost unconstrained merger of regions of a size smaller than a predefined threshold with neighboring, larger regions. Adopting this approach could help improve the algorithm's performance w.r.t. e_r^m for some video sequences by reducing over-segmentation tendencies. For some other sequences (e.g. larger moving object followed by a significantly smaller one, cp. "Container Ship"), however, the other measures would be degraded.

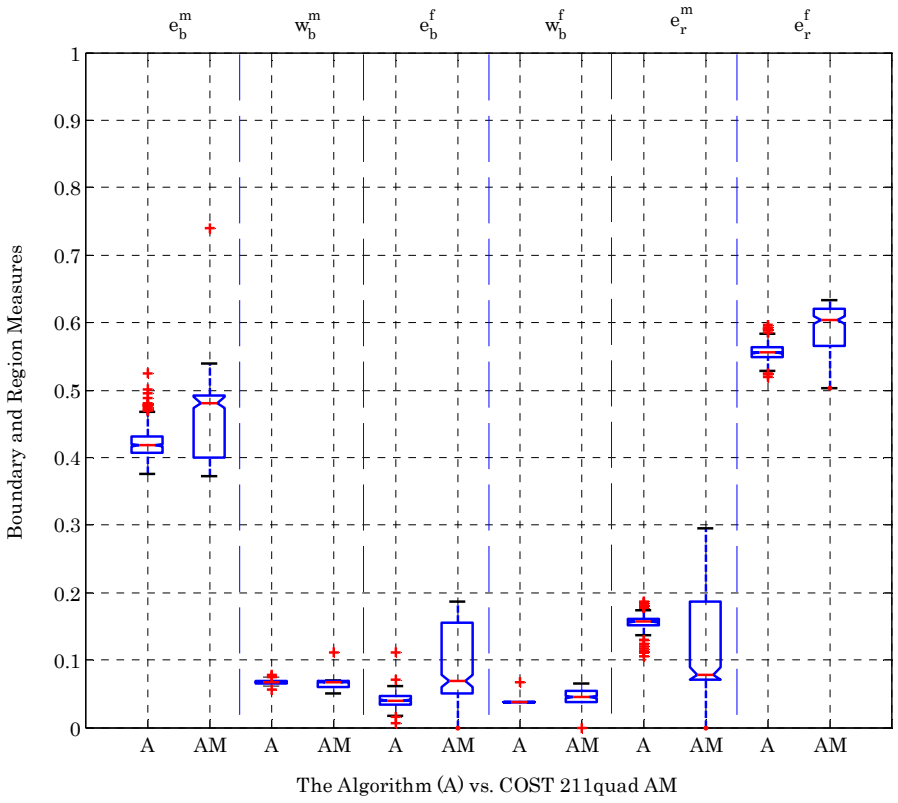


Fig. 29 – Boxplots of Huang and Dom's measures for the "Container Ship" sequence

Although the evaluations yield an overall favorable outcome for the algorithm, it can not be ignored that some of the absolute error rates can be seen as relatively high. In fact, segmentation is one of the most difficult tasks in an image processing chain. For that, segmentation errors must be expected and handled. Thus in the content-based video coding framework presented in this thesis, this problem is tackled by providing video quality assessment tools for identification of decoded video quality impairments (cp. Sec. 3.3).

4.5 Discussion

The spatio-temporal texture analysis algorithm proposed in this thesis has been successfully designed to meet the requirements of the analysis-synthesis video coding framework. The analyzer can, however, be further improved. Special care has to be taken of the over-segmentation tendency of the proposed method and the minimization of the absolute error rates. This may be achieved by inserting additional features into the proposed framework as a long-term segment trajectory observation criterion for improved merger decisions. An automatic weighting of spatial and temporal hints appears to be a further important approach to avoid a priori prioritization of these features. Finally, constrained tracking based on explicit modeling of the temporal consistency of label assignment is required in order to further improve the analysis efficiency.

5 Texture Synthesizer

As a result of the texture analysis process, a mask sequence showing the synthesizable parts, i.e. the detail-irrelevant textures, of the considered GoP is generated. In this chapter, a detailed description of the post-processing and usage of the segmentation masks will be given. Two texture synthesis approaches, applicable to different texture classes, will further be presented. A preliminary description of the state-of-the-art in texture synthesis will be given in the following section.

5.1 Previous Work

Texture synthesis is an important problem in the field of computer graphics and computer vision. It consists in generating a synthetic texture that is objectively different from but perceptually similar to a given texture example. That is, the synthetic texture should be subjectively perceived as being generated by the same underlying stochastic process as the texture source. Hence, texture synthesis approaches are designed to emulate the generative process of the texture they are aiming to reproduce and typically stretch. As the texture sample is usually much smaller than the synthetic texture to generate, simple verbatim tiling of the former normally yields subjectively annoying edges and repetitions. For that, algorithms that are more sophisticated are required to tackle the texture synthesis problem with good prospects. Typical applications of texture synthesis algorithms are special effects for cinema and television, computer-generated animation, computer games [141], and video restoration [142]. Some texture synthesis algorithms are also used within the framework of texture recognition and identification for validation of corresponding analysis methods.

The major problems that have to be tackled in any texture synthesis process are roughly two-fold. The first challenge relates to the proper estimation of the underlying stochastic process of a given texture based on a finite sample of it. The second task refers to the formulation of an efficient sampling procedure to generate new textures from a sample [143]. The former challenge steers the accuracy of the synthetic textures, while the latter, also referred to as probability density function (pdf) sampling, determines the computational complexity of the

texture generation procedure. Clearly, the texture synthesis problem is ill-posed as the texture sample could have been drawn from an infinite number of different textures. Hence, additional assumptions have to be made in order to further constrain the problem. Typically, it is assumed that the sample is "large enough" to capture the stationarity of the texture and that the scale of the texture elements is known [144].

Texture synthesis approaches can be divided into two categories: Parametric and non-parametric. In both synthesis categories, the pdf of a given texture example is approximated and sampled to generate new, perceptually similar texture samples as already said above.

Parametric synthesis approaches approximate the pdf of the texture source using a compact model with a fixed parameter set [26],[39],[145],[146],[147],[148]. Such approaches entail helpful hints w.r.t. the understanding of the underlying texture properties, which can be helpful in identification and recognition scenarios.

Non-parametric synthesis approaches do not explicitly estimate the pdf by which the texture is assumed to be modeled, they rather measure it from the texture example, which can be a 2D image or a video signal. Non-parametric approaches typically formulate the texture synthesis problem based on the Markov Random Field (MRF) theory [149],[144],[143],[150],[151],[152],[153],[154],[155]. The MRF model is characterized by statistical interrelations within local vicinities. That is, each texture sample or texture sub-set is predictable from a small set of spatially neighboring samples and is independent of the rest of the texture. The underlying generative stochastic process of the texture source is assumed to be both local and stationary in the MRF context. Non-parametric approaches can be pixel or patch-based. Pixel-based algorithms update the synthetic texture pixelwise [149],[144],[143], while patch-based approaches operate a patchwise update [150],[151],[152],[153],[154],[155], i.e. a set of samples is updated simultaneously. Not only do non-parametric synthesis approaches yield better synthesis results than parametric algorithms, also can they be successfully applied to a much larger variety of textures [154].

An important drawback of MRF approaches resides in their limited capacity to capture texture structure. For instance are low frequency interactions between samples difficult to represent as MRF techniques only consider local dependencies between pixels, whereas the former interactions relate to long range effects (both spatially and temporally). Existing methods, parametric and non-parametric, moreover tend to generate artifacts given small scale structure or object boundaries [156],[155].

In terms of compactness of texture representation, non-parametric synthesis algorithms are typically less efficient than parametric ones. Thus, in the content-based video coding framework, parametric synthesis methods are usually used [22]. It will be shown, however, that non-parametric synthesizers can be used with a compact side information, which is yet achieved by transferring higher computational complexity to the decoder. The synthesis scenario presented in this thesis corresponds to a texture generation process with boundary conditions, which can also be referred to as inpainting of video sequences.

Note that the principle of the parametric texture synthesis algorithms described in this chapter bears some resemblance to model-based video coding schemes (cp. Sec. 2.3). While the latter are confined to a restricted synthesis scenario (e.g. head and shoulder scenes), the former are typically applicable to a larger set of (background) textures.

5.2 Synthesis of Rigid Textures

The texture synthesizer presented in this section is designed to synthesize rigid video textures. The underlying hypothesis of its conception is that rigid textures typically undergo global motion that can be explained by either camera operations or the self-motion of the corresponding foreground or background object. Assuming that texture analysis has generated reliable segmentation masks, the challenge then consists in modeling the global motion of the given rigid texture in a compact manner. The texture synthesizer presented in this section has an undeniable similarity to GMC approaches [1],[28],[78]. It can, however, be viewed as a generalization of these. In fact can this texture synthesizer account for several moving objects in a video sequence, while GMC

approaches typically compensate the camera motion and thus capture only one of the possibly multiple homogeneously displaced regions [28],[78].

5.2.1 GoP Structure

The incoming video sequence is divided into overlapping groups of pictures (GoP). The first GoP consists of the first picture of the sequence and the last picture of the GoP can a priori be set at an arbitrary distance to the first picture of the GoP. The two pictures are used as key pictures (K) by the texture synthesizer for rigid textures. Between the key pictures, an arbitrary number of pictures, also called synth pictures and denoted S in the following, are partially or fully synthesized depending on their content. For example, when 3 synth pictures are used, the first GoP has the structure K_0SSSK_1 in temporal order. The second GoP consists of the last picture of the first GoP (the K_1 picture) and the next key picture. That is, the second GoP has the structure K_1SSSK_2. The n^{th} GoP will thus have the structure $K_{n-1}SSSK_n$. Note that the number of synth pictures can in principle vary from GoP to GoP, which accounts for specific motion content in the given video sequence.

5.2.2 Post-Processing of Texture Analysis Masks

The texture analysis module, presented in Sec. 4.2.4, provides the masks showing the textures to be synthesized. These masks must be post-processed to fulfil the prerequisites of the texture synthesizer. The post-processing step examines the adequacy of a given key picture to provide a corresponding synthesis picture with a given missing texture. The post-processing operation thereby consists in matching the texture regions in the synthesis pictures with the corresponding ones in the key pictures. This is achieved by warping the identified texture regions in the current picture towards the corresponding textures in the key pictures. Warping is done using the planar perspective motion model (30). Notice that warping, i.e. motion estimation, is done based on Newton's method. The application of the latter to motion estimation has been extensively described by Smolić [78]. Within the current system, rigid texture synthesis can only be performed with valid samples if the texture to be replaced in the current picture can be found in one of the key pictures. Fig. 30 depicts the post-processing procedure. The hatched regions correspond to textures of the same class in this figure, independently of the kind of hachure.

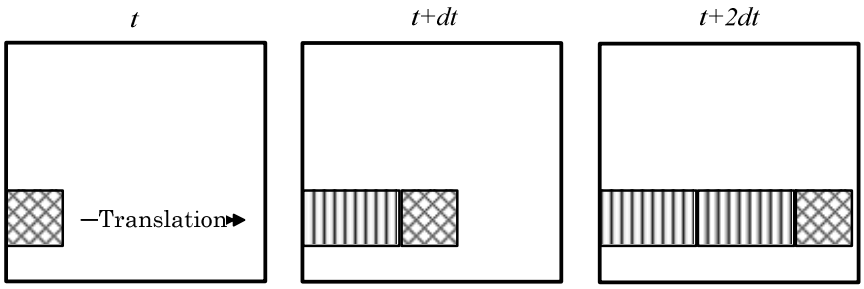


Fig. 30 – Post-processing of segmentation masks. Segmentation mask of a key picture showing a hatched region of a given class (left). Segmentation mask of current picture showing a hatched region of the same class with uncovered areas (middle), segmentation mask of a key picture showing a hatched region with larger uncovered areas (right).

Only the diagonally hatched area depicted at time instance $t + dt$ can be provided by the reference picture at time instance t, while both diagonally and vertically hatched areas are available in the key picture at time instance $t + 2dt$. Hence, the size of the synthesized region in the current picture at time instance $t + dt$ can only be maximized if both key pictures are considered simultaneously. It can however occur that some samples of the current synthetic picture are not available in either reference picture. In that case, a non-empty set of invalid samples exists in the synthetic area. These samples can either be discarded from synthesis or not. This decision must be taken at the encoder and signaled to the decoder as side information. Discarding texture areas from synthesis, however, yields shrunk segmentation masks, which implies reduced bit rate gain expectations compared to the genuine codec.

The warping operations conducted in Fig. 30 may reveal that no good motion estimation results can be achieved given the key pictures. Hence, no synthesis can be operated under such circumstances. In the current picture, only parts of the texture region that yield good motion estimation results are maintained after post-processing, while the remainder are discarded. The "goodness" of the motion estimation result is measured based on the distortion criterion proposed by Smolić [78], which is the mean squared error. A threshold is used in this thesis to discriminate good and bad predictions.

5.2.3 Performing Synthesis

5.2.3.1 Valid Samples

The synthesizer for rigid textures warps the missing texture from the key picture with the best match towards a given synthesizable texture region[1] identified by the texture analyzer and post-processed as described in Sec. 5.2.2. The remaining, valid, unsynthesized samples are replaced using the second reference picture (cp. Fig. 31).

5.2.3.2 Invalid Samples

As already explained in Sec. 5.2.2, after the valid samples have been replaced, some unsythesized samples may remain. They are tackled by using a Markovian intra-synthesis approach, where the remaining "holes" are filled-in by assigning them corresponding color components of the defined sample that features the most similar neighborhood properties in the given picture and lies within a limited search range. A 3x3 neighborhood and an 11x11-search range are used.

5.2.4 Side Information

The texture synthesizer for rigid textures requires some side information to infer the synthetic textures from the available texture examples in the key pictures. These meta data mainly correspond to a segmentation mask, two motion parameter sets, and a control parameter for each synthesizable texture region identified by the texture analyzer. Exhaustive side information semantics are given in Tab. 3.

The segmentation mask indicates, which sample subset of the current picture corresponds to a region to be synthesized. The former also indicates if and where the textures identified in the synth pictures can be found in the corresponding key pictures (cp. Fig. 30). A binary mask is transmitted to the decoder. The control parameter signals the key picture priorities given a texture to be synthesized. The perspective motion (30) of the given texture region is estimated

[1] The principle of the texture synthesis module for rigid textures was developed in collaboration with my former colleague Bela Makai. The early implementations of this synthesizer were done by Bela and referred to its operability proof. The author further improved the algorithm with regard to post-processing of the segmentation masks, the bidirectionality of the approach, and its H.264/MPEG4-AVC integration including the side information definition.

w.r.t. the first and the last picture of the GoP as described in [157]. The key picture that yields the motion parameters $(a_1,\ldots,a_8)$ with the smallest difference signal between synthesized and original texture region is assigned the highest priority. The second reference picture is considered for synthesis if and only if it also yields a good motion prediction. Two motion parameter sets, one per key picture, are transmitted as side information in that case. More generally, up to one motion parameter set is transmitted per key picture. The motion parameters are uniformly quantized and their quantization step size can be varied. It must also be signaled to the decoder, how invalid samples are to be handled (cp. Secs. 5.2.2 and 5.2.3). That is, whether they should be discarded from synthesis at the cost of coding (bit rate) efficiency, or whether they should be kept at the eventual cost of low synthesis quality.

Additional parameters are transmitted if a single pair of motion parameter sets is to be used for several non-connected regions. These parameters are referred to as "split" parameters in this thesis and comprise a "split" flag signaling that a given synth picture contains at least two non-connected textures that can be described by a common pair of motion parameter sets.

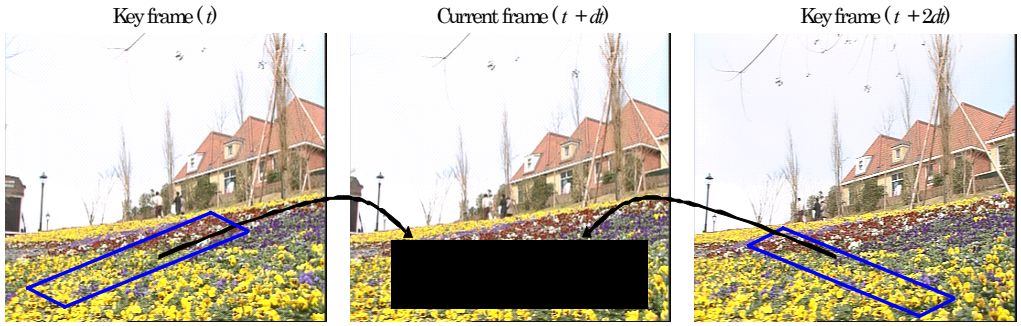


Fig. 31 – Warping of texture from key pictures towards region to be filled

Is the "split" flag set, so must the labels of the texture regions in the current mask be transmitted for the synthesizer to identify the "split regions". Fig. 32 depicts an example where "split" regions occur. It is assumed here that the analysis algorithm yields two segments within the flowerbed and one within the sky texture. As the flowerbed features no local motion activities, both corresponding segments can be described by a common pair of motion parameter sets.

Fig. 32 – Semantic of "Split" parameters

This is signaled to the synthesizer by setting the "split" flag to one and by transmitting the order of occurrence of the labels in the segmentation mask, i.e. 1,2,2. Where the texture regions are ordered by scanning the binary mask from top to bottom and from left to right starting at the upper left corner. Note that a texture region can also be split due to the post-processing operation described in Sec. 5.2.2. A burst marker is finally transmitted for each picture and indicates to which picture type (synth or ref) the given picture corresponds.

Side info.	Amount	Occurence	Semantics
Mask	1	Picture	Segmentation mask with highlighted synthesizable texture regions
Control Flag	1	Segment	Key picture with highest priority
$(a_1,\ldots,a_8)$	≤ 2	Segment	Perspective motion parameters (30) for warping
Split Flag	1	Picture	Motion parameters usable for more than one texture segment ?
Split Order	X	Picture	Labels of texture segments in segmentation mask. Transmitted only if "Split Flag" set
Burst Marker	1	Picture	Indicator for picture's affiliation to ref or synth picture
isDiscarded	1	Segment	Flag for signaling handling of invalid samples
$(QP_1^{synth},\ldots, QP_8^{synth})$	1	Sequence	Quantization parameter set for motion parameters

Tab. 3 – Side information for rigid texture synthesis

This synthesizer works remarkably well for rigid objects with the assumptions implied in the motion model (30). As the proposed synthesizer has limited degrees of freedom, it is not individually evaluated in this section. Extensive experiments are conducted in Chapter 10 within the overall framework.

5.3 Synthesis of Non-rigid Textures

The texture synthesis framework by Kwatra et al. [154] is used in this thesis as it represents a non-parametric approach with the ability to synthesize a broad range of texture varieties. In fact, plane (2D) as well as volumetric (3D) textures (e.g. fire, smoke, water etc.) can be successfully rendered with this approach. Optimization potentialities of the framework by Kwatra et al. will be explored in the following. Some fundamentals will, however, be introduced first for better legibility of their approach.

5.3.1 Fundamentals

The approach by Kwatra et al. [154] is based on Markov random fields (MRF). That is, an infinite texture sample T is defined as a homogeneous random field with the aim to generate a synthetic texture from it. The Markovianity of the random field thereby relates to

$$\mathrm{P}\left(\mathbf{t}_i \middle| T - \{i\}\right) = \mathrm{P}\left(\mathbf{t}_i \middle| N_i\right), \tag{69}$$

that is to the assumption that any texture pattern $\mathbf{t}_i$ extracted from the given sample T at location i can be predicted from the corresponding neighborhood system N_i and is independent of the rest of the texture. The homogeneity property presumes that the conditional probability $\mathrm{P}(\mathbf{t}_i | N_i)$ is independent of the site i in T. MRF allow modeling of non-linear statistical dependencies between amplitude values, e.g. luminance, in a given neighborhood system [4].

A formal definition of MRF-based texture synthesis can now be given as follows. Let L stand for an image or video to be synthesized, let $\mathbf{t}_i$ be an unknown sub-area or sub-volume of L, and let $\mathbf{t}_i^N \subset L$ stand for the amplitude pattern in the given neighborhood system N of $\mathbf{t}_i$. The implementation of MRF then requires the estimation of the conditional probability $\mathrm{P}(\mathbf{t}_i | \mathbf{t}_i^N)$. This is a formidable task as the number of $\mathbf{t}_i^N$ constellations is considerable. For instance, let the number of amplitude levels be 256 and let $\mathbf{t}_i$ represent a single pixel. Further assuming a neighborhood of four pixels yields a total of 256^4 possible constellations of $\mathbf{t}_i^N$. Typically, in the texture synthesis framework, texture examples feature only a

sub-set of the possible $\mathbf{t}_i^N$ constellations. Hence, some of the latter are never considered in the course of the synthesis process.

In the non-parametric texture synthesis framework the given texture sample is assumed to capture the statistics of the underlying infinite texture, which is an ill-posed problem as already discussed above. In this framework, the conditional probability distribution function $P(\mathbf{t}_i \mid \mathbf{t}_i^N)$ is approximated and then sampled. For that, a perceptual similarity measure s is needed. Assuming an infinite texture sample T, the set

$$M = \left\{ \mathbf{t}_i' \subset T \;\middle|\; \left\| \mathbf{t}_i^N - \mathbf{t}_i^{N'} \right\| \approx 0 \right\} \tag{70}$$

refers to the sub-textures $\mathbf{t}_i'$ of T with a perceptually similar neighborhood to the unknown sample $\mathbf{t}_i$. They can be seen as samples of the conditional pdf of $\mathbf{t}_i$. The selection of the most adequate pdf sample is subject to constraints with regard to the given application. Typically, pdf sampling is done with regard to the similarity criterion $\| \; \|$ [144] that steers the quality of the synthesis results. In practice, a finite texture sample T' is usually given. Hence, M is approximated by M', which implies that no matches may be found in T' given a specific neighborhood system N [144].

As explained above, MRF approaches describe visual patterns by imposing statistical constraints and learn models directly from the signal. Hence, they are also called synthesis by example approaches.

5.3.2 Graph Cut

Network flow algorithms have been studied in the field of combinatorial optimization [158]. They have been initially used to solve problems in physical networks. Network flow approaches have yet been abstracted and extended to problems with no physical network. In fact do a number of computer vision tasks consist in constrained label assignment to image samples (cp. Sec. 4.1, Sec. 4.2.1, Sec. 4.2.3, and Sec. 4.2.4) that can be conveyed into an energy minimization framework. The latter can in turn be represented as a network flow problem. Hence have network flow algorithms found widespread application in optimization problems that can be represented by graphs containing nodes and arcs between these nodes [159],[160]. For instance, given an adequate graph configuration and defined edge capacities, image restoration, image segmentation or disparity estimation problems can be tackled. The first application of graph cuts to the texture synthesis framework has been proposed by Kwatra et al. [154].

5.3.2.1 Principles

A flow network $G(Nd,E)$ is composed of nodes Nd and edges E with corresponding weights $K(E)$. The latter can also be seen as edge capacities. The nodes typically correspond to single pixels or a set of pixels in the computer vision context. Two of the vertices (nodes) with special attributes are called the source $src \in Nd$ and the sink $snk \in Nd$. src and snk are usually seen as the class labels that can be assigned to the nodes in vision. Such a flow network builds the foundation of the graph cut problem. The source node features only outgoing edges, while the sink node possesses only incoming ones. The network thus features a flow from src to snk . In case of a fully connected graph, the two edges between two normal vertices are directed and can feature deviating capacities, which is required in some graph-based applications. Fig. 33 depicts a flow network or graph, where the (asymmetric) capacities of the edges are given by their respective strength. Note that the source and the sink are also called terminal nodes in the literature [161] and that a graph can contain more than two terminal nodes. In this work, symmetric weights are assumed.

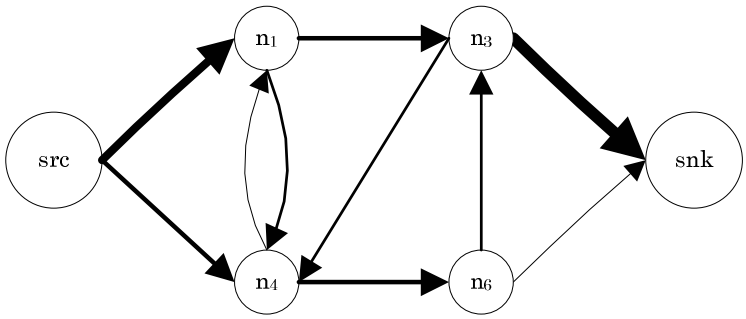


Fig. 33 – Schematic example of a flow network

The minimum cut problem consists in partitioning the graph $G(Nd,E)$ into two disjoint subsets $T_{src} \subset Nd$ and $T_{snk} \subset Nd$, with $T_{src} \cap T_{snk} = \emptyset$, $src \in T_{src}$, and $snk \in T_{snk}$, such that the sum of the capacities of the cut edges is minimized. That is, if E_{cut} is the set of the cut edges, then the following expression is to be minimized

$$\mathrm{E} = \sum_{p=1}^{|E_{cut}|} \mathrm{w}(e_p) \text{ with } \mathrm{w}(e_p) \in K(E) \ \wedge \ e_p \in E_{cut}. \tag{71}$$

According to the theorem by Ford and Fulkerson, solving the min-cut problem is equivalent to solving the max-flow determination task [162]. Hence, the set of min-cut edges, E_{cut}, is determined by finding the maximum flow through the network. For better illustration, the graph can de seen as a set of nodes linked by directed pipelines with capacities corresponding to the edge weights. Now, pumping as much water as possible into the source and observing the amount of water reaching the sink gives the maximum flow of the network. Some pipelines will be saturated in this experiment, i.e. they convey as much water as possible given the pipeline's capacity. Hence, the network's nodes will be divided into two disjoint subsets. The flow $\mathrm{f}(e)$ through an edge can be formalized as a function that gives the number of units the edge can convey (72).

$$\mathrm{f}: Nd \times Nd \rightarrow \mathrm{IR} \tag{72}$$

where IR is the set of real values. The network flow is now constrained by three properties, i.e. the positivity constraint

$$\forall (n, p) \in Nd \times Nd \ , \ \mathrm{f}(e_{np}) \geq 0 \tag{73}$$

and the capacity constraint

$$\forall (n, p) \in Nd \times Nd \ , \ \mathrm{f}(e_{np}) \leq \mathrm{w}(e_{np}) \tag{74}$$

stating that the flow through the pipeline e_{np} from node n to node p can not exceed its capacity $\mathrm{w}(e_{np})$. The third requirement, the flow conservation constraint, relates to Kirschhoff's law. It states that the total flow into a node is identical to the total flow leaving the node and vice versa. That is, a node can not store the incoming flow. The flow conservation constraint applies to all vertices of the graph except src and snk. The total flow towards node p can be given as

$$\mathrm{f}_{\mathrm{in}}(e_p) = \sum_{n=1}^{|Nd|} \mathrm{f}(e_{np}) . \tag{75}$$

The total flow leaving node p can be given as

$$\mathrm{f}_{\mathrm{out}}(e_p) = \sum_{n=1}^{|Nd|} \mathrm{f}(e_{pn}) . \tag{76}$$

The flow conservation constraint can be formulated as

$$\forall \ p \in Nd - \{src, snk\}, \mathrm{f}_{\mathrm{in}}(e_p) = \mathrm{f}_{\mathrm{out}}(e_p) \ . \tag{77}$$

Given the above-mentioned constraints, the max-flow through a cut can be given as

$$\mathrm{f}(e_{src,snk}) = \sum_{n \in T_{src}}^{|E_{cut}|} \sum_{p \in T_{snk}}^{|E_{cut}|} \mathrm{f}(e_{np}) \ . \tag{78}$$

Based on the properties of the network flow presented above, approaches for the max-flow and min-cut determination can be developed.

5.3.2.2 Max-Flow and Min-Cut Algorithms

Single-source, single-sink min-cut or max-flow algorithms can typically be subdivided into two main classes. Some approaches are based on the Ford-Fulkerson method. Other approaches are based on the "push-relabel" technique. A thorough discussion of these approaches is beyond the scope of this work. The reader is referred to [162],[163],[161],[159] for more details. Notice that the algorithm proposed by Kolmogorov [161] is used in this work as recommended in [160].

5.3.3 Video Synthesis using Graph Cuts

The synthesis algorithm developed by Kwatra et al. [154] can a priori be applied to plane and volumetric textures. It is non-parametric and can thus render a large variety of video textures. An overview of their approach is given in Fig. 34. The synthetic texture is updated patchwise by matching its defined neighborhoods with the given texture sample and by placing (patch placement) the identified continuation patches in an overlapping manner.

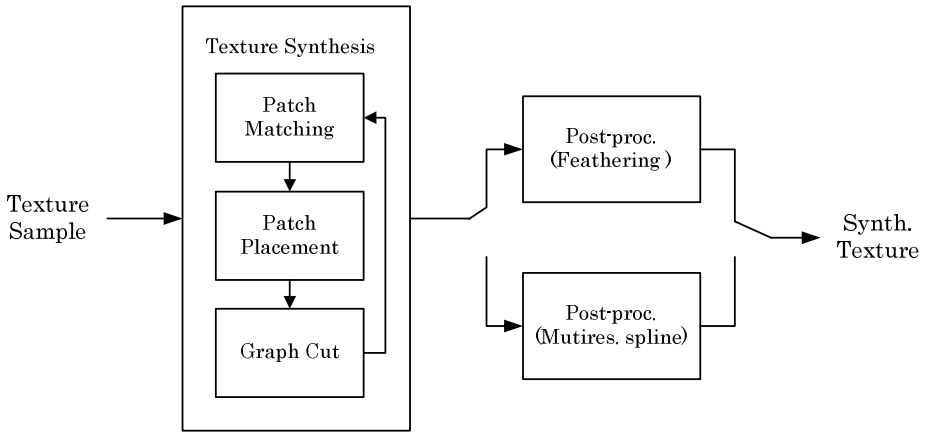


Fig. 34 – Overview of the texture synthesis algorithm by Kwatra et al. [154]

The originality of Kwatra et al.'s approach resides in the fact that they formulate the texture synthesis problem as a graph cut issue. Hence, the optimal seam between two overlapping patches, i.e. the seam yielding the best possible MRF likelihood among all possible seams for the given overlap, can be computed. This approach minimizes subjectively annoying edges at patch transitions [154]. Post-

processing tools are provided in case of subsistence of subjectively annoying edges.

5.3.3.1 Sampling Procedure

Given a texture sample T' and an empty lattice L to be filled in, Kwatra et al. define three sampling approaches. They all have in common that only unused offsets in the texture sample are considered to determine continuation patches for the output texture. This is done to avoid noticeable periodicity in the synthetic texture.

The first procedure is called the random placement by the authors [154]. The entire texture sample is placed in L at random offsets and in an overlapping manner. This approach is very fast and can be used for stochastic textures, i.e. textures with no dominant oriented structures. The blending procedure in overlap regions will be explained in the next section.

The second procedure is the entire patch matching. It follows the same idea as the random placement method. The difference between both approaches lies in the fact that, here, the offsets in the texture sample are obtained through matching. That is, the output texture is initialized with the texture sample at a random offset. The subsequent offsets of the sample are determined under consideration of the cost function

$$\mathrm{E}_{\mathrm{ep}} = \left\| \mathbf{t}_i - \mathbf{l}_{i+j} \right\| \tag{79}$$

where $\mathbf{l}_{i+j}$ corresponds to the portion of translated output (translation j) overlapping the input [154]. This cost function basically corresponds to the normalized sum of squared errors, where the normalization factor is the size of the overlap region. The selected offset is determined with regard to the following stochastic criterion

$$P \sim \mathrm{e}^{-\frac{\mathrm{E}}{k\sigma^2}} \tag{80}$$

where σ corresponds to the standard deviation of the texture sample's amplitudes. k represents a degree of freedom that steers the randomness of the

offset selection. The lower the choice of k, the more does the offset selection correspond to minimizing (79). The higher k, the more is the minimization constraint relaxed. The entire patch matching method can be used for structured or semi-structured textures as their inherent periodicity is captured by the cost function above [154].

The third procedure is the sub-patch matching approach. It is comparable to the entire patch matching method. Here, patches of a predefined size, typically much smaller than the texture sample, are disposed in $\boldsymbol{L}$. The first sub-patch can thereby be selected at random.

$$E_{sp} = \left\| \mathbf{t}_{i-j} - \mathbf{l}_i \right\| \tag{81}$$

The costs for a given translation j of the input texture's sub-patch (at location i, overlap region) in $\boldsymbol{T}'$ are given in (81). Offset selection is operated as given in (80). The sub-patch matching approach is the most effective of the three sampling methods. It can be applied to stochastic and volumetric textures. It captures the local coherency of spatially unstructured textures like water, smoke, etc.. The size of the sub-patch, copied from the texture sample towards $\boldsymbol{L}$, is chosen in a way that it is slightly larger than the overlap region in the output texture. This is done to ensure that the output texture is grown with each update patch. Note that the summed term in (79) and (81) can be implemented by subtracting the pixel colors. No color space is, however, recommended by Kwatra et al. [154]. The HSV color space is used in this thesis due to its perceptual uniformity.

The matching approaches presented above can be computationally very expensive, especially when 3D textures like fire, water or smoke are synthesized. Hence, Kwatra et al. use an FFT-based acceleration algorithm that can be applied to sum of squared error cost functions. Details with regard to this run-time optimization can be found in [154].

5.3.3.2 Graph Cut Formulation of Texture Synthesis

As already explained above, the synthesized texture L is grown patchwise, where the current update patch is placed in an overlapping manner in the output texture. Kwatra et al. [154] propose a graph cut formulation of the problem of finding an adequate seam between overlapping patches. Once the overlap region (synthetic texture) and the continuation patch (texture sample) have been found, the graph cut algorithm determines the path that minimizes the subjective annoyance of the blending within the overlap region. Fig. 35 depicts this problem based on a 2D texture synthesis example. The path defines which irregular shaped portion of the continuation patch (patch B), found in the texture sample, must be transferred to the synthetic texture. Due to the irregular shape of the copied region, blocking effects can be avoided and seamless transitions generated given "good" continuation patches. Potentially subjectively annoying artifacts of the blending are captured by an adequate cost function that is applied to every pixel transition in the overlap region. The cost function used by Kwatra et al. [154] is defined as

$$\mathrm{E}_{\mathrm{cut}}(s,p,A,B)=\frac{\mathrm{N}_{\mathrm{cut}}(s,p,A,B)}{\mathrm{D}_{\mathrm{cut}}(s,p,A,B)}=\frac{\left\|\mathbf{a}(s)-\mathbf{b}(s)\right\|+\left\|\mathbf{a}(p)-\mathbf{b}(p)\right\|}{\left\|\delta_{\mathrm{a}}(s,\vartheta)\right\|+\left\|\delta_{\mathrm{b}}(s,\vartheta)\right\|+\left\|\delta_{\mathrm{a}}(p,\vartheta)\right\|+\left\|\delta_{\mathrm{b}}(p,\vartheta)\right\|} \quad (82)$$

where $\mathbf{a}$ corresponds to the overlap region in patch A, while $\mathbf{b}$ is the overlap region in patch B. Note that $\mathbf{a}$ and $\mathbf{b}$ are considered as vectors here for better legibility. s and p are two adjacent pixels in the overlap region, while $\delta_{\mathrm{a,b}}(p,\vartheta)$ represents the gradient at location s or p in direction ϑ. $\|\ \|$ constitutes an adequate norm. The cost function constrains the optimal path determined by the graph cut algorithm. Hence, its formulation is crucial with regard to the quality of the synthesis results.

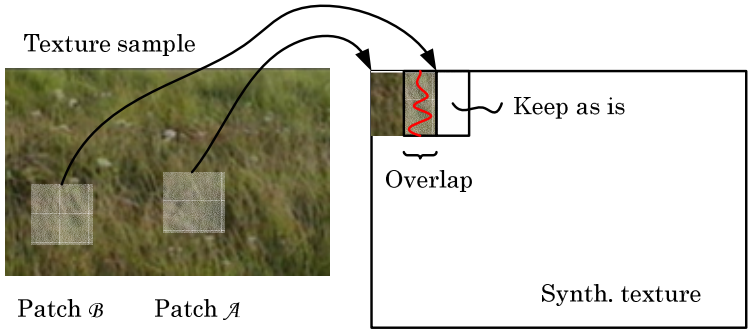


Fig. 35 – Assembly of two patches with a seam determined by a graph cut algorithm

The graph cut formulation of the texture synthesis problem is depicted in Fig. 36. A 5x5 grid is shown, where each of the numbered square boxes corresponds to a pixel in the overlap area. A may be seen as the patch to be continued in the output texture, while B would represent the continuation patch found in the example texture T'. The graph cut algorithm links adjacent pixel pairs via the cost function defined in (82).

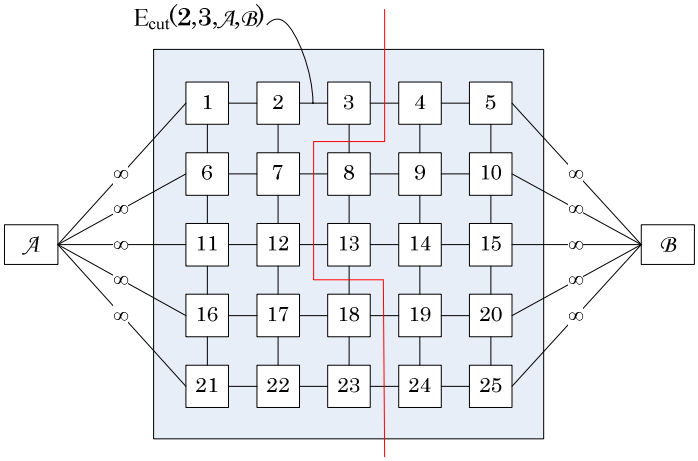


Fig. 36 – Graph cut principle given a texture synthesis scenario

Let A stand for the source and B for the sink. Contiguous pixels to nodes A and B are then linked to the latter with infinite weights. Hence, a cut at these transitions is made impossible, as it would yield infinite costs. This is done in order to constrain samples adjacent to sink and source to come from B and A respectively, which reflects the fact that false boundaries at transitions between overlap region and terminal nodes should be avoided. The optimal cut, i.e. the cut yielding minimum costs, is determined by applying the optimization algorithms

mentioned in Sec. 5.3.2. The former is depicted in Fig. 36 as a red line. This cut specifies the contribution of each patch to the overlap region. For instance, in Fig. 36, the pixels at the left hand side of the cut are provided by patch A, while the pixels at the right hand side are provided by patch B.

Efros and Freeman [152] proposed a texture synthesis algorithm based on image quilting. Their approach is patch-based and the synthetic texture is also grown with overlapping patches. They propose a similar blending approach called minimum error boundary cut, which is based on dynamic programming. Kwatra et al. argue that the graph-cut formulation of the problem is more generic as old seams can be taken into account, which can not be done using dynamic programming [154].

5.3.3.3 Consideration of Old Seams

The originality of Kwatra et al.'s approach can be seen in its ability to incorporate seams generated by previous cut operations into the current minimum cut determination problem. A new seam, at possibly reduced costs, can then be determined. This problem is relevant in the sense that it arises a number of times during the synthesis process. The challenge now resides in the proper integration of old seam costs into a graph. In fact, it must be ensured that the old seams are either improved or at least remain unchanged. Therefore do Kwatra et al. insert so-called seam nodes at old seam locations. A one dimensional illustration of this problem is given in Fig. 37. Let A and C be overlapping patches. Assume that there exists an old seam between node (pixel) s_A and node p_B, where s_A was formerly taken from patch A and p_B came from patch B in a previous graph cut operation. Let A represent the pixels already available in the output texture, while C is the continuation patch, found in the texture sample, that is to be copied to the ouput texture (cp. Fig. 38).

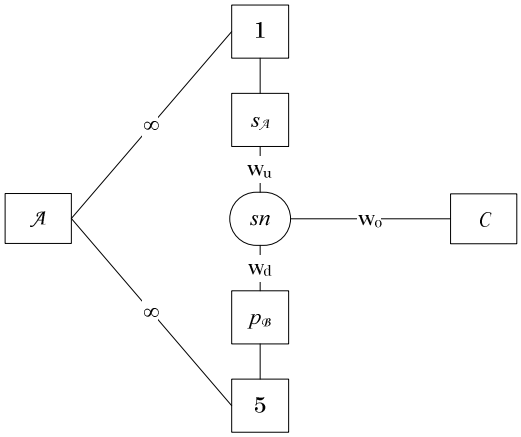


Fig. 37 – Accounting for old seams in the current minimum cut determination

A seam node (sn) is inserted between nodes s_A and p_B. It is linked to the neighboring nodes with the costs w_u and w_d respectively. w_u and w_d thereby correspond to $E_{cut}(s_A, p_C, A, C)$ and $E_{cut}(s_C, p_B, C, B)$ respectively. sn is also linked to C with the cost w_o that corresponds to $E_{cut}(s_A, p_B, A, B)$, which in turn is the old seam cost. In Fig. 37, it can be seen that, if the edge between patch C and the seam node is cut, the latter remains in the synthetic texture. On the other hand, if the edge between node s_A and the seam node is cut, p_B remains linked to C and must be updated with p_C. A cut between sn and p_B yields the update of s_A with s_C. If an edge between the seam node and one of the adjacent pixels is cut, the new seam is considered in the final cost of the min-cut. The equivalence between old seam costs improvement and the min-cut of the graph is given only if the considered cost function is a metric [154]. That is, E_{cut} must satisfy

$$\begin{aligned} &E_{cut}(s,p) = 0 \Leftrightarrow s = p \\ &E_{cut}(s,p) = E_{cut}(p,s) \geq 0 \\ &E_{cut}(p,q) \leq E_{cut}(p,s) + E_{cut}(s,q) \end{aligned} \tag{83}$$

where the so-called triangle inequality, applied to the old seam problem, states that the cost of any edge originating from a seam node must be smaller than the sum of the costs of the two other edges. Hence, if E_{cut} is a metric, it is guaranteed that at most one of the edges originating from a seam node is cut. Notice that, the

cut can theoretically be done such that none of the edges originating from sn in Fig. 37 is affected. In such cases, the old seam is overwritten, i.e. s_A and p_B, and does not contribute to the final costs of the cut. Note that the patch designations have been ignored in (83) for better legibility.

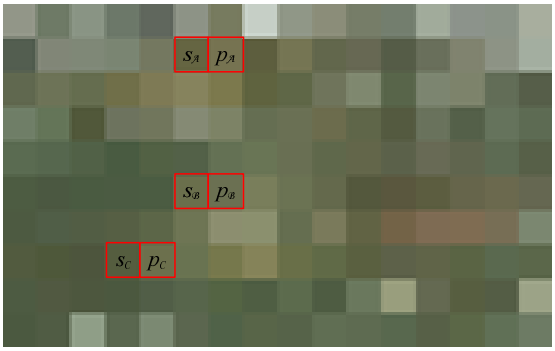


Fig. 38 – Locations A, B, and C, in the texture sample, that can be used for texture synthesis

Kwatra et al. also use the approach described above for cases where the output texture has already been totally filled. Seams featuring high costs can then be refined by matching the surrounding region with the texture sample. The selected match can then be placed in the output texture such that the critical seam is totally covered. The border pixels of the overlap region are constrained to come from the synthetic texture in order to avoid visible seams. The graph is then built and the minimum cut determined as described above.

As can be seen, Kwatra et al.'s old seam handling potentially allows the improvement of old seams a posteriori when a new patch is placed at the corresponding locations. Their graph cut formulation is derived from but not equivalent to the work of Boykov et al. with respect to the α-expansion algorithm [160]. The latter allows any set of image pixels to change their labels to α [160], where α corresponds to C in the example depicted in Fig. 37. Notice that the mathematical basement for Kwatra et al.'s graph architecture is given in [164].

5.3.3.4 Accounting for Subjectively Annoying Seams

It has been argued that the graph cut formulation of the texture synthesis problem yields optimal seams between adjacent patches. As an overlap region can encompass several old seams in practice, no adequate continuation patch may be available in the texture sample T'. For that, the optimal seam may still be visible and thus subjectively annoying.

Kwatra et al. tackle this problem by using feathering techniques around the seams. That is, the seams are smoothed by applying a low-pass filter, typically a Gaussian kernel, in their vicinity [154]. Kwatra et al. report that, depending on the type of texture, multiresolution splining may be a better alternative to feathering. Multiresolution splining, introduced by Burt and Adelson [165], can be used to smooth "very obvious" seams. This approach, however, has the drawback that the contrast of the output image may be reduced, when a number of small patches have been placed in the output texture. Kwatra et al. [154] manually choose between feathering and multiresolution splining on a case to case basis.

5.3.3.5 Video Synthesis

The approach proposed by Kwatra et al. can be used for 2D still image as well as 3D video texture synthesis. In this work, only video synthesis will, however, be discussed. The reader is referred to [154] for specialized implementations of Kwatra et al.'s algorithm with regard to 2D texture synthesis.

The extension of the graph cut formulation of the texture synthesis problem from 2D to 3D is straightforward. For the 3D synthesis, the video sequence is seen as a volume composed of voxels (volume elements). The patches are spatio-temporal cuboids that can be placed anywhere in the synthesized texture volume. The min-cut can be seen as a surface within the 3D space. Kwatra et al. propose three video synthesis scenarios that depend on the properties of the given video sequence [154]. They all have in common that they aim to infinitely loop the input video, i.e. the synthetic texture is an infinite temporal extension of the original one. In two of the scenarios, the input video sequence is assumed to be temporally stationary, while spatio-temporal stationarity is assumed in the third case. In the former case, the patch selection procedure can be restricted to the

temporal axis, as no good matches can be expected in the spatial dimensions. Given spatio-temporal stationarity, all space and time dimensions are explored for continuation patch detection. The different scenarios will be presented into more detail in the following. Note that, the approach by Kwatra et al. does not encompass a stationarity analysis. The latter is basically done subjectively or based on the trial-and-error method.

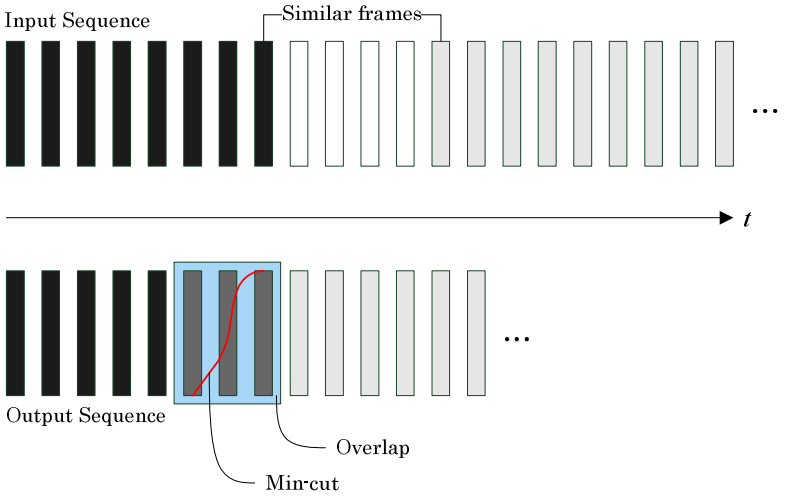


Fig. 39 – Seamless extension of a temporally stationary video sequence

The first scenario consists in a constrained seam detection for video transitions. Is a given video texture temporally stationary, then can a temporal seam be determined by operating a pairwise comparison of the corresponding pictures. Has an adequate seam been detected, so can a graph be built within a spatio-temporal window around that seam (cp. Fig. 39). The spatio-temporal min-cut is thereby determined according to the approach described above (cp. Fig. 36). Hence, the optimal temporal transition is individually determined for every pixel in the transition window. By repeating such transitions, the video sequence can be infinitely looped [154]. The input video sequence should, however, be "long enough" to avoid obvious repetitions.

The second scenario is called the random temporal patch offsets approach [154]. It is mainly applied to short video clips for which an adequate seam may not be found by pairwise picture comparison. This approach also takes into account that, in such clips, repetitions are easily noticeable. In this scenario, input

texture patches are placed at random offsets in the synthetic texture volume. The seam is then computed within the whole spatio-temporal volume of the input texture [154].

The third scenario consists in the synthesis of spatio-temporally stationary video sequences. For the latter, during patch selection, both temporal and spatial dimensions are explored. For that, the sub-patch matching procedure is used, which yields improved results. These ameliorations relate to the fact that some video textures feature a dominant motion that can not be captured by simple picture to picture comparison [154]. Thus, the seam detection approach used in the first scenario can not be applied here. Kwatra et al. propose an alternative approach for looping spatio-temporally stationary sequences. Here, the same N_f^{synth} pictures of the input video are placed at the beginning and at the end of the output texture. These pictures are constrained to remain unmodified via infinite costs in the graph (cp. Fig. 36). The output texture is then filled between the two groups of length N_f^{synth} under consideration of this constraint. Once the output texture has been filled, the first N_f^{synth} input pictures are removed from it. Infinite looping is guaranteed by this approach as the picture number N_f^{synth} in the output texture is identical to the last picture of the output before the removal operation. Note that the input video sequence can not only be extended along the time axis, but also along the spatial dimensions if spatial translations are allowed in the synthetic texture volume.

5.3.3.6 Limitations of the Algorithm

Kwatra et al. [154] do not prove that solving the min-cut problem is equivalent to obtaining optimal labeling of the graph including seam nodes. This gap is filled by Qin and Yang [164], who gave the mathematical proof of the optimality of label assignments using the graph cut formulation described above.

An important prerequisite for the usage of the old seam optimization is that the cost function is a metric as already explained above. The cost function defined by Kwatra et al. [154] is given in (82). The hypothesis that their cost function is a metric is tested in the following using a numerical example.

Let $\mathbf{a}(s) = (2,7,9)$, $\mathbf{b}(s) = (2,7,1)$, $\mathbf{a}(p) = (1,2,0)$ and $\mathbf{b}(p) = (3,3,0)$, where any 3D color space (RGB, HSV, etc.) is assumed here. $\mathrm{E}_{\mathrm{cut}}(s,p,A,B)$ can then be given as

$$\mathrm{E}_{\mathrm{cut}}(s,p,A,B) = \frac{8+3}{0+1+6+0} = \frac{11}{7} \tag{84}$$

where the ℓ_1 norm is used and the gradients have been directly inserted into (84). Let $\mathbf{c}(\mathrm{s}) = (2,7,1)$ and $\mathbf{c}(\mathrm{p}) = (3,8,0)$. $\mathrm{E}_{\mathrm{cut}}(s_A,p_C,B,C)$ and $\mathrm{E}_{\mathrm{cut}}(s,p,A,C)$ can then be given as

$$\mathrm{E}_{\mathrm{cut}}(s_A,p_C,B,C) = \frac{0+5}{6+0+0+1} = \frac{5}{7} \tag{85}$$

and

$$\mathrm{E}_{\mathrm{cut}}(s,p,A,C) = \frac{8+8}{0+1+0+1} = \frac{16}{2} = 8 \tag{86}$$

where, again, the gradients have been directly inserted into (85) and (86). It can be seen from (84), (85) and (86) that

$$\mathrm{E}_{\mathrm{cut}}(s,p,A,C) \geq \mathrm{E}_{\mathrm{cut}}(s,p,A,B) + \mathrm{E}_{\mathrm{cut}}(s_A,p_C,B,C) \tag{87}$$

which contradicts the triangle inequality in (83). Same conclusion can be drawn, when ℓ_2 is used instead of ℓ_1. Hence, the cost function defined in (82) is not a

metric, such that the old seam optimization proposed by Kwatra et al. can not be operated reliably.

5.3.4 Proposed Improvements to Graph Cut Synthesis

The graph cuts video synthesis approach proposed by Kwatra et al. [154] is non-parametric and patch-based. It corresponds to one of the best synthesis algorithms available to date. For this reason is their texture synthesis framework used in this thesis. In this section, some optimization proposals to Kwatra et al.'s approach are introduced. They relate to the cost function (82) as well as extensions aiming conformance to generic video synthesis and coding requirements.

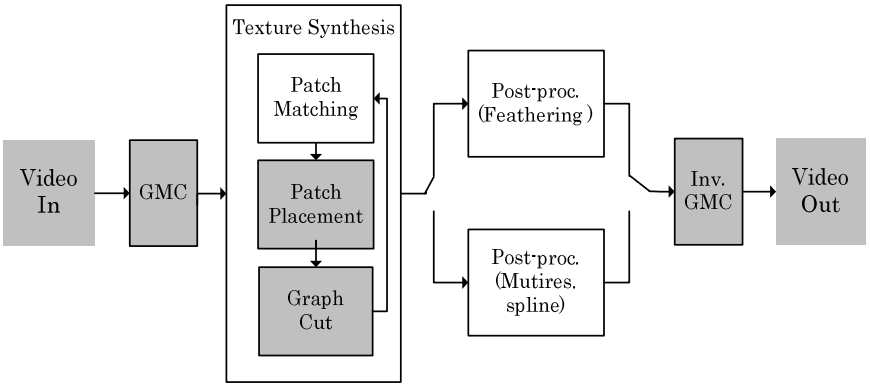


Fig. 40 – Overview of modified modules compared the texture synthesis algorithm by Kwatra et al. [154]

They involve the specification of the required GoP structure, the consideration of spatio-temporal constraints, the formulation of the required global motion compensation (GMC) for generic video synthesis, as well as the definition of the required side information. The modified or added modules can be seen in Fig. 40, where they are highlighted with a gray shade.

5.3.4.1 GoP Structure

In the constrained synthesis scenario, the input video sequence is temporally segmented as depicted in Fig. 41. The first group of pictures consists of a reference burst (R1) that temporally precedes the synthetic burst (S1). The synthetic burst is itself followed by another reference burst (R2) in temporal order. The two reference bursts and the synthetic burst give a group of bursts

(GoB) R1S1R2. The reference bursts are chosen such that they contain the sample texture T' required to synthesize the empty lattice L in the synthetic burst. The second GoB consists of the last reference burst of the first GoB, R2, and the next synthetic (S2) and reference (R3) bursts to give R2S2R3. Hence, an overlapping GoB structure is used. The succeeding GOBs are composed accordingly until the end of the video sequence is reached.

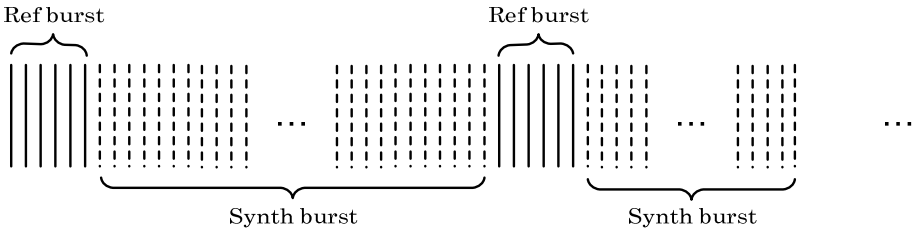


Fig. 41 – GoP structure of the texture synthesizer for non-rigid textures

Note that the GoP structure described above and depicted in Fig. 41 exhibits some similarity with the video synthesis approach for spatio-temporally stationary textures described by Kwatra et al. [154] and summarized above. The difference between the two GoP structures resides in the fact that no repetition of reference bursts is done in the structure presented in this thesis. The sub-patches used are also placed in the synth burst such that they temporally overlap the reference bursts, which is not the case in Kwatra et al.'s [154] work.

5.3.4.2 Spatio-Temporal Constraints

Kwatra et al. [154] assume that the segmentation problem is solved. That is, their approach applies to a boundary-condition-free context where the required texture samples are assumed to be given. The video coding framework considered in this thesis requires the extension of Kwatra et al.'s algorithm to a texture synthesis module with boundary constraint handling. The challenge thus resides in the transposition of a given (unconstrained) textures synthesis algorithm onto an inpainting scenario (cp. Sec. 5.2, Fig. 31), where missing textures, that can be thought of as spatio-temporal "holes" in a given video sequence, must be filled. This is a somewhat complicated task as both spatial and temporal inferences are required. Inappropriate synthesizer decisions may yield annoying artifacts as flickering or spurious spatio-temporal edges. Several inpainting approaches have been proposed in the literature [166],[167],[142],[168]. They often relate to filling

in relatively small holes in a still image or a video sequence and typically require user-interaction to identify synthesizable regions. Inpainting algorithms are, for instance, required in image or video restoration applications, where image or film material degradations can occur in the form of line scratches. In order to tackle larger "holes" in an inpainting framework, several hybrid algorithms have been proposed in the literature [155],[169]. They typically combine the advantages of texture synthesis and inpainting. Such approaches are usually limited to still images or correspond to a straightforward extension of 2D approaches, where each picture of a given video sequence is handled separately. An very interesting inpainting approach for object removal that exploits spatio-temporal video properties has been recently proposed by Patwardhan et al. [170].

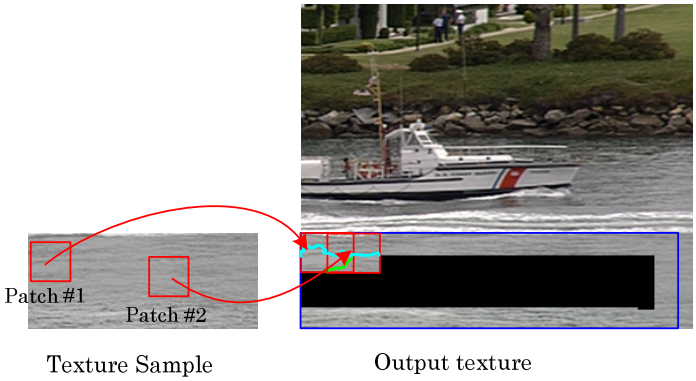


Fig. 42 – Constrained texture synthesis principle

Given the constraint synthesis scenario, the patch placement procedure is affected (cp. Sec. 5.3.3). In fact, due to the constraint boundaries, the first patch can not be selected at random in texture sample T'. The definition of an adequate patch placement approach is however straightforward. The patches at the boundary of the synthesis area must be placed such that they overlap the boundary texture (cp. Fig. 42). This in turn implies that the constraint texture must be the same as the texture to be synthesized, which must be determined by the texture analysis module (cp. Fig. 42). The graph cut algorithm is now applied to aforesaid overlap region, which yields an irregular boundary between the spatio-temporal constraint region and the synthetic video texture. This ideally decreases the perceptibility of the boundary given an adequate cost function.

Fig. 43 – Progressive fade-in of non-rigid, synthetic texture (white samples)

Irregular boundaries are also obtained in temporal direction. These are depicted in Fig. 43, where masks showing the synthetic (white) and original (black) pixels in a partially synthesized video sequence are given. It can be seen that a temporal, progressive fade-in of the synthetic texture is achieved. That is, the proportion of synthetic samples continuously grows in a transition phase between ref and synth burst until the whole region to be synthesized is filled with synthetic samples. Ideally, the fade-in occurs seamlessly. The same effect takes place at the transition from synth the ref burst in the shape of a fade-out process of the synthetic texture.

5.3.4.3 Monitoring the Filling Process

Bit masks are defined in this thesis to handle generic synthetic region shapes. The masks also allow monitoring of the progress of the synthesis algorithm, of the state of the pixels in the region to be synthesized as well as of those in the example texture.

The texture sample's bit mask has four state bits. They are LOCK, INVALID, FREE and USED (cp. Fig. 44). The LOCK state is assigned to pixels that do not belong to the texture under consideration. Due to implementation constraints, the texture sample is typically held in a cuboidal volume, even if the former is of arbitrary shape. The samples belonging to the synthetic texture must thus be distinguished from the others via the LOCK state. All pixels that lie within a half patch of the LOCK area are marked INVALID. Such pixels correspond to the boundary condition of the texture to be synthesized and can not be used as patch centers while sampling the texture example. Using an INVALID sample as patch center would yield patches with LOCK samples. USED pixels correspond to texture example locations that have already been sampled. Such locations and a

corresponding neighborhood can be locked to avoid obvious repetitions in the synthetic texture. FREE samples are pixel locations that have not been sampled. In cases where no FREE samples are available in the texture example, the matching criterion is simply minimized without considering the LOCK labels. Fig. 44 depicts a bit mask corresponding to a texture sample T'. i thereby represents a valid location for placement of the center of a continuation patch.

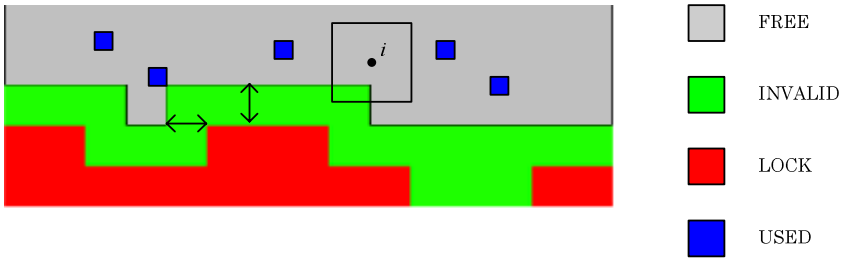


Fig. 44 – Bit mask for monitoring the texture sample state

The output bit mask, i.e. the state mask relating to the synthetic texture, features four state bits. That is the EMPTY, the ORIG, the SYNTH, and the CONST bits (cp. Fig. 45). The EMPTY state indicates where the pixels to be synthesized can be found in the ouput sequence. The CONST bit is set at locations within the boundary condition. It must be guaranteed by the texture analyzer that the identified non-rigid texture is surrounded by a corresponding boundary condition. The width of the latter constitutes a degree of freedom of the present texture synthesis approach and must be determined according to the selected patch size. CONST pixels can be replaced in the course of the synthesis process in order to obtain irregular transitions between synthetic texture and boundary condition (cp. Fig. 45). CONST samples are also used for matching, i.e. for determination of continuation patches in the texture example T'. ORIG pixels correspond to locations that are not modified by the synthesis algorithm and typically contain textures that differ from the texture to be synthesized. SYNTH pixels correspond to samples that have been copied from the texture example towards the texture to be synthesized. The shape of the SYNTH region of a given continuation patch is determined by the graph cut algorithm.

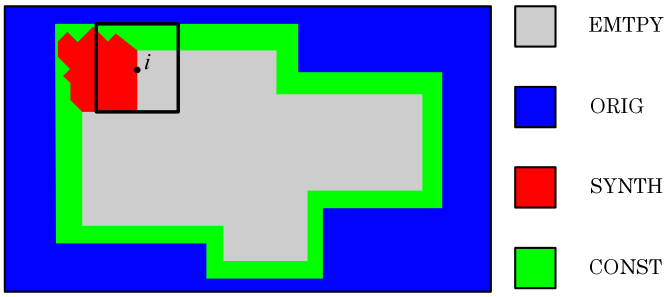


Fig. 45 – Bit mask for monitoring the output texture state

Fig. 45 depicts a bit mask corresponding to a texture area to be synthesized. *i* thereby corresponds to a valid location for placement of the center of a continuation patch. Notice that the synthetic texture region can be of arbitrary shape in the present implementation.

The index mask holds the location of each pixel of the synthetic texture in the texture sample. The index mask thus has the same resolution as the output texture. This mask is required for several operations in the course of the synthesis process. The former is used to determine patch transitions by locating index discontinuities. The latter are used to determine transition costs for instance. The index mask is also used for post-processing operations as feathering (cp. Sec. 5.3.3).

5.3.4.4 Filling Strategies

The filling order of the 3D, i.e. spatio-temporal "hole", can have a significant impact on the quality of the synthesis result [155]. As already noticed above, the synthesis is done in a constrained environment in this work. That is, a spatio-temporal extrapolation of the given, temporally preceding texture sample is operated and must, at some stage, be conformed to the temporally subsequent texture sample. It must thus be ensured that special precautions are taken at the transitions between natural and synthetic textures. An intuitive approach may consist in growing the synthetic texture from one temporal end to the other (cp. Fig. 46). This may yield discontinuities (unnatural motion) at the second temporal end of the considered texture.

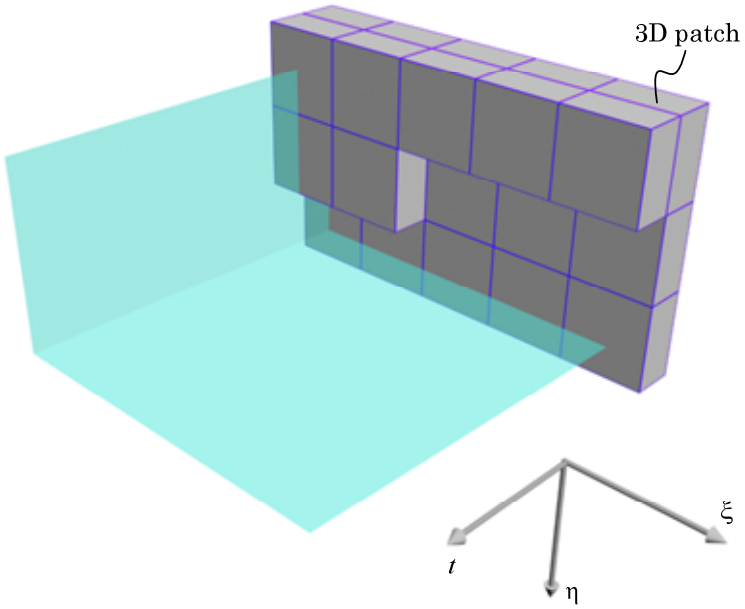


Fig. 46 – “Scanline” filling procedure

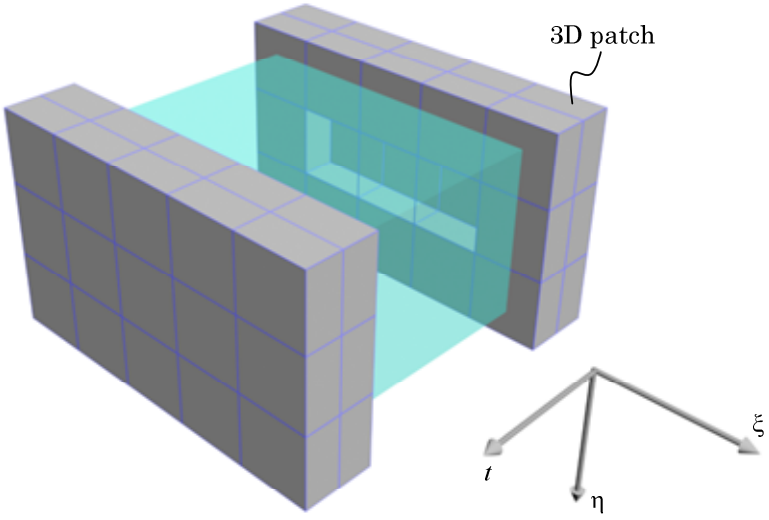


Fig. 47 – Helical filling procedure

In case of annoying subjective artifacts, this effect can be attenuated by ensuring a minimum overlap between the extrapolated patches and the constraint texture, which allows construction of sufficiently complex graphs such that min-cut optimization potentialities with respect to the cut path are given. However may temporal inconsistencies be still easily noticeable. An alternative filling strategy consists in displacing the conversion area of the synthetic texture towards to middle of the corresponding volume. This complies with a helical filling strategy (cp. Fig. 47). Inconsistencies due to merging extrapolated textures growing in different spatio-temporal directions is now "hidden" within the synthetic texture, which often improves subjective perception of the latter. Notice that, in Fig. 46 and Fig. 47, the 3D patches are represented as gray cuboids. Both approaches only differ in the patch placement chronology as already explained above. Fig. 46 depicts the "scanline" filling method, where the filling order is $\xi,\eta,t,\xi,\eta,\ldots$ In contrast, the helical filling approach, shown in Fig. 47, is characterized by the filling order $\xi,\eta,t,t,\ldots,\xi,\eta,t,\ldots$ This filling order is, however, not respected in Fig. 47 for legibility reasons.

5.3.4.5 Cost Function

The cost function used by Kwatra et al. is given by (82). The distance between the color components of two adjacent pixels s and p is determined in the numerator $\mathrm{N}_{\mathrm{cut}}(s,p,A,B)$. The denominator corresponds to gradients in the neighborhood of s and p. It is assumed that, due to visual masking effects, cuts should be placed in texture areas showing high gradients, where these seams are expected to be less visible.

Kwatra et al. do not recommend any color space for the implementation of the cost function. Due to its perceptual uniformity (cp. Sec. 4.1.3), the HSV color space is used in this thesis. The cost function is thus reformulated as follows

$$\begin{aligned}\mathrm{N}'_{\mathrm{cut}}(s,p,A,B) = {} & \mathrm{d}_{\mathrm{ph}}\left(H_{\mathbf{a}(s)},H_{\mathbf{b}(s)}\right)+\mathrm{d}_{\mathrm{ph}}\left(H_{\mathbf{a}(p)},H_{\mathbf{b}(p)}\right)\\ & +\left\|S_{\mathbf{a}(s)}-S_{\mathbf{b}(s)}\right\|+\left\|S_{\mathbf{a}(p)}-S_{\mathbf{b}(p)}\right\| \\ & +\left\|V_{\mathbf{a}(s)}-V_{\mathbf{b}(s)}\right\|+\left\|V_{\mathbf{a}(p)}-V_{\mathbf{b}(p)}\right\|\end{aligned} \quad . \tag{88}$$

The ℓ_1 norm is used for the saturation and value components. It must be accounted for the fact that the hue component represents an angle. Hence must another norm

$$\mathrm{d}_{\mathrm{ph}}\left(H_{\mathbf{a}(s)},H_{\mathbf{b}(s)}\right)=\min\left(\left\|\mathbf{a}_H(s)-\mathbf{b}_H(s)\right\|,2\pi-\left\|\mathbf{a}_H(s)-\mathbf{b}_H(s)\right\|\right) \tag{89}$$

where $\|\ \|$ again corresponds to the ℓ_1 norm.

As already discussed above, the cost function defined in (82) is not a metric, such that old seam optimization is not used in this thesis. This choice is motivated by the fact that the usage of a metric (e.g. the numerator of (82)) in combination with old seam optimization typically yields relatively regular cuts that in turn yield noticeable spatial and temporal artifacts.

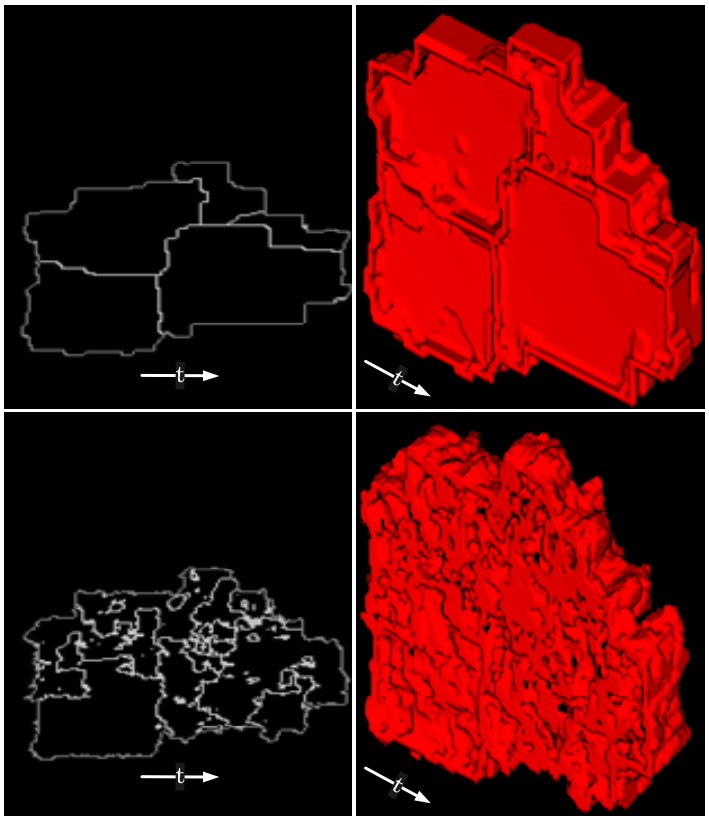


Fig. 48 – Graph cut texture synthesis with a metrical and a non-metrical cost function. 2D min-cuts obtained for the metrical (top left) and the non-metrical (bottom left) cost functions, 3D min-cuts obtained for the metrical (top right) and the non-metrical (bottom right) cost functions.

Fig. 48 depicts the effects caused by metrical functions in the constrained synthesis framework (top row). It can be seen that especially the temporal cuts are mostly planar. This corresponds to an abrupt transition between neighboring patches, which typically yields jerky motion. A non-metrical cost function typically yields significant improvements of the min-cut properties as can be seen in the bottom row of Fig. 48, where spatial and temporal transitions are more gradual.

5.3.4.6 Temporal Alignment

The System of Kwatra et al. implicitly assumes a static camera scenario [154]. This is a very restrictive framework, as many natural video sequences feature some degree of camera motion. This constraint has to be relaxed for achieving a generic texture synthesis tool for content-based video coding.

Camera motion is typically not known a priori and requires a motion estimation process. The outcome of the latter can then be used for global motion compensation, also called temporal alignment in the following. Note that, for better legibility, the continuous spatial coordinates (x, y) will be used indifferently to address continuous and discrete coordinates in this section. Fig. 49 depicts the general principle of the temporal alignment problem considering the "Coast Guard" test sequence as an example. Note that the regions that can be used as reference to follow the explanations below are marked with a bounding box.

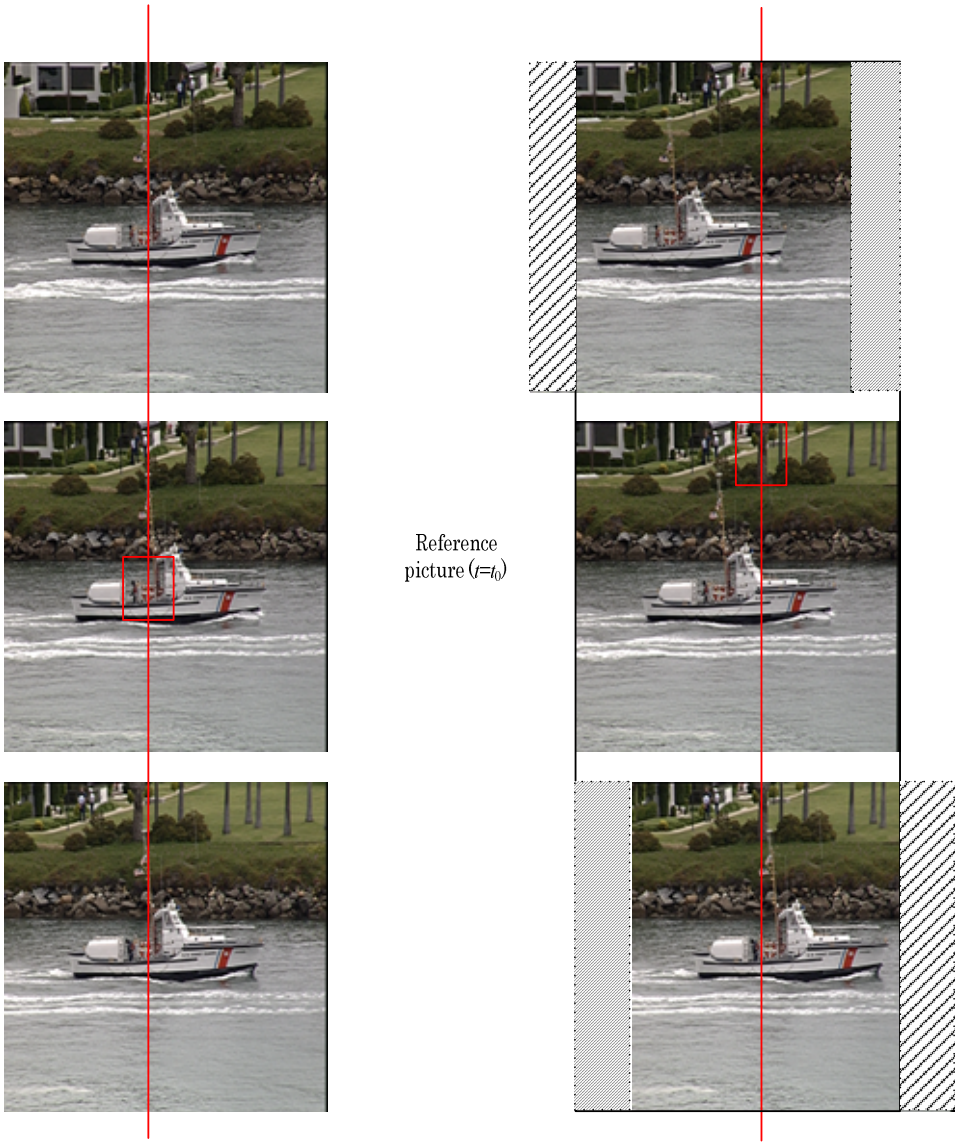


Fig. 49 – Implications of the temporal alignment operation

The left column corresponds to three pictures of the test sequence that features a translational global camera motion from left to right. The pictures are 30 frames or 1s apart, the upmost one being the first to occur in temporal order. It can be seen in the original video sequence (left column in Fig. 49), that the background, e.g. around the middle column of the pictures, is moving due to camera motion. Defining a reference time instance t_0 and aligning the temporally neighboring pictures to it yields the result depicted in the right column in Fig. 49. Alignment corresponds to warping (simple translation in this example) the upper and the lower pictures towards the reference picture such that the background appears to

be static. Ideally, the motions of the water and the ship are thereby preserved. Note that the hatched areas correspond to covered or uncovered regions in the aligned pictures. These regions cannot be synthesized by the present approach.

Once the temporal alignment has been operated w.r.t. the reference time instant t_0, the assumptions of Kwatra et al. [154] with respect to camera motion ideally hold. Applying the proposed synthesis method yet results in a synthetic texture w.r.t. time instance t_0. Back-warping the synthetic pictures towards the genuine time instant yields a total of two interpolation steps per sample (warp and back-warp), which may give blurry results. Hence, the number of interpolations is minimized for improved visual quality by operating "virtual synthesis" in the warped domain. That is, each sample (texture samples, constraint and synthetic regions) is assigned a unique index within a group of bursts in the genuine coordinate system. Warping is applied both to the video signal and to the index maps yielding a first 3D matrix of spatio-temporal index maps, $\mathbf{M}$, in the warped domain. The synthetic samples are inserted into the warped video, while their indexes are inserted into a second matrix of index maps, $\mathbf{M'}$, during synthesis. Finally, the synthetic samples, of which the indexes are held by $\mathbf{M'}$, are assigned to the unwarped region to be synthesized by looking up their destination in $\mathbf{M}$ at the same spatio-temporal location. Note that the back-warping step is omitted in Fig. 40 for legibility purposes.

An approach for robust global motion compensation has been implemented in this thesis. It is based on the M-estimation algorithm presented in Sec. 4.2.4. Fig. 50 depicts a picture burst that is to be temporally aligned. The first step of the temporal alignment algorithm now consists in determining the perspective motion model (30) between adjacent pictures starting from the outmost pictures. Note that the picture-to-picture motion estimation is symbolized by the dashed arrows in Fig. 50.

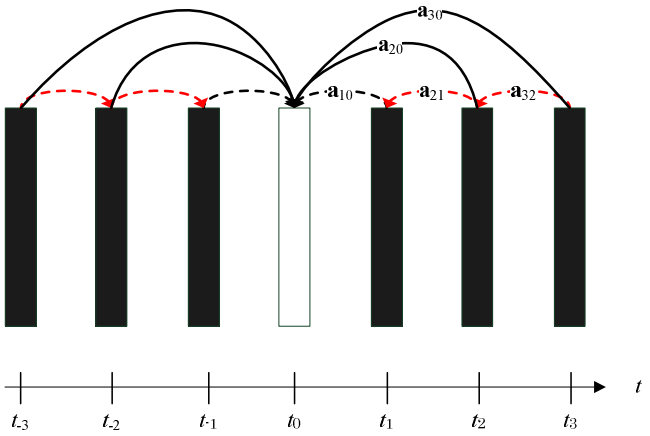


Fig. 50 – Temporal alignment of a picture burst

Once the picture-to-picture global motion is known, the reference time instance t_0 is shifted towards the designated picture, e.g. the mid-burst picture, by accumulation of the motion parameter sets. This is represented by the solid arrows in Fig. 50. The accumulation can be obtained by chaining single, i.e. picture to picture, perspective (30) transformations. Let (x, y) be a sample location in the current picture, further let (x', y') be the corresponding, warped sample location in the temporally succeeding picture. Finally, let (x'', y'') be the warped sample location of (x', y') in the corresponding temporal successor. If $\varphi_8^x(x,y)$ and $\varphi_8^y(x,y)$ are defined as the perspective transformations in x and y directions, then the following equations apply

$$x' = \varphi_8^x(x,y) = \frac{a_1 + a_3 x + a_4 y}{1 + a_7 x + a_8 y}$$
$$y' = \varphi_8^y(x,y) = \frac{a_2 + a_5 x + a_6 y}{1 + a_7 x + a_8 y} \tag{90}$$

$$x'' = \varphi_8^x(x',y') = \frac{a_1' + a_3' x' + a_4' y'}{1 + a_7' x' + a_8' y'}$$
$$y'' = \varphi_8^y(x',y') = \frac{a_2' + a_5' x' + a_6' y'}{1 + a_7' x' + a_8' y'} \tag{91}$$

Determining the accumulated motion parameters now corresponds to determine $\hat{\varphi}_8^x(x,y)$ and $\hat{\varphi}_8^y(x,y)$ such that

$$x'' = \hat{\varphi}_8^{\,x}(x,y)$$
$$y'' = \hat{\varphi}_8^{\,y}(x,y) \quad . \tag{92}$$

The inference of $\hat{\varphi}_8^{\,x}(x,y)$ and $\hat{\varphi}_8^{\,y}(x,y)$ from (30) is straightforward and can be initiated through insertion of (90) into (91) as follows

$$x'' = \frac{a_1' + a_3' a_1 + a_4' a_2 + (a_1' a_7 + a_3' a_3 + a_4' a_5)x + (a_1' a_8 + a_3' a_4 + a_4' a_6)y}{1 + a_7' a_1 + a_8' a_2 + (a_7 + a_7' a_3 + a_8' a_5)x + (a_8 + a_7' a_4 + a_8' a_6)y}$$
$$y'' = \frac{a_2' + a_5' a_1 + a_6' a_2 + (a_2' a_7 + a_5' a_3 + a_6' a_5)x + (a_2' a_8 + a_5' a_4 + a_6' a_6)y}{1 + a_7' a_1 + a_8' a_2 + (a_7 + a_7' a_3 + a_8' a_5)x + (a_8 + a_7' a_4 + a_8' a_6)y} \tag{93}$$

The genuine perspective transformation formulation defined in (30) can be obtained by normalizing (93) by

$$\alpha_0 = 1 + a_7' a_1 + a_8' a_2 \tag{94}$$

which yields

$$x'' = \frac{\alpha_1 + \alpha_3 x + \alpha_4 y}{1 + \alpha_7 x + \alpha_8 y}$$
$$y'' = \frac{\alpha_2 + \alpha_5 x + \alpha_6 y}{1 + \alpha_7 x + \alpha_8 y} \tag{95}$$

with

$$\alpha_1 = \frac{a_1' + a_3' a_1 + a_4' a_2}{1 + a_7' a_1 + a_8' a_2}$$
$$\alpha_2 = \frac{a_2' + a_5' a_1 + a_6' a_2}{1 + a_7' a_1 + a_8' a_2}$$
$$\alpha_3 = \frac{a_1' a_7 + a_3' a_3 + a_4' a_5}{1 + a_7' a_1 + a_8' a_2}$$
$$\alpha_4 = \frac{a_1' a_8 + a_3' a_4 + a_4' a_6}{1 + a_7' a_1 + a_8' a_2}$$
$$\alpha_5 = \frac{a_2' a_7 + a_5' a_3 + a_6' a_5}{1 + a_7' a_1 + a_8' a_2}$$
$$\alpha_6 = \frac{a_2' a_8 + a_5' a_4 + a_6' a_6}{1 + a_7' a_1 + a_8' a_2}$$
$$\alpha_7 = \frac{a_7 + a_7' a_3 + a_8' a_5}{1 + a_7' a_1 + a_8' a_2}$$
$$\alpha_8 = \frac{a_8 + a_7' a_4 + a_8' a_6}{1 + a_7' a_1 + a_8' a_2} \quad . \tag{96}$$

(95) and (96) can now be applied recursively to estimate large global displacements between two pictures of arbitrary temporal distance from each other. This is however limited by estimation and modeling inaccuracies that accumulate with the number of successive recursions [78]. Notice that, in this thesis, the temporal alignment algorithm is applied on a full GoB (cp. Fig. 41).

5.3.4.7 Side Information

The side information required by the texture synthesizer for non-rigid textures, to infer the synthetic textures from the available texture samples in the reference bursts, mainly corresponds to a segmentation mask and a motion parameter set for each picture of the video sequence to be synthesized.

The segmentation mask indicates, which sample subset of the current picture corresponds to a region to be synthesized. The former also indicates if and where the textures identified in the synth pictures can be found in the corresponding reference pictures. Note that a binary mask is transmitted. The perspective global motion parameters (30), $(a_1,\ldots,a_8)$, of each picture with respect to the reference time instance are estimated as described above. These motion parameters are transmitted to the synthesizer as side information. They are uniformly quantized and their quantization step size can be varied. A burst marker that is also transmitted for each picture indicates to which burst type (synth or ref) the given picture corresponds.

Additional parameters are transmitted if several non-connected regions of the same class are available in the same mask. These parameters are referred to as "split" parameters in this thesis and comprise a "split" flag signaling that a given synth picture contains at least two non-connected textures of the same class. Is the "split" flag set, so must the labels of the texture regions in the current mask be transmitted for the synthesizer to identify the "split" regions. The required side information for the texture synthesizer described in this section is summarized in Tab. 4.

Side info.	Amount	Occurrence	Semantics
Mask	1	Picture	Segmentation mask with highlighted synthesizable texture regions
$(a_1,\ldots,a_8)$	≤ 1	Picture	Perspective motion parameters (30) for temporal alignment
Split Flag	1	Picture	More than one region of the same class in mask ? Order of texture regions changed ?
Split Order	X	Picture	Labels of texture segments in segmentation mask. Transmitted only if "Split Flag" set.
Burst Marker	1	Picture	Indicator for picture's affiliation to ref or synth burst
$(QP_1^{synth},\ldots,$ $QP_8^{synth})$	1	Sequence	Quantization parameter set for motion parameters

Tab. 4 – Side information for non-rigid texture synthesis

Note that the synthesis method by Kwatra et al. [154] can a priori be used to synthesize rigid textures. Nonetheless has the texture synthesizer presented in Sec. 5.2 proved to be very effective for such textures as its side information (including example textures) is significantly more compact than the one required for the synthesizer for non-rigid textures.

5.3.5 Experimental Results

In this section, experimental evaluations are conducted to demonstrate that the extensions to Kwatra's approach [154], proposed in this thesis, entail significant perceptual gains in the constrained texture synthesis framework.

5.3.5.1 Ground Truth Set

The ground truth set selected for the evaluations consists of five video sequences with different variations of the water texture. The latter texture is considered in the experiments because it is a good representative of the class of non-rigid textures. The selected video clips are extracted from the "Whale Show", the "Talking Man", the "Synchronized Swimming", the "Life Belt", and the "Sea" sequences. These clips feature different lighting conditions (e.g. daytime, nighttime) and properties (e.g. small waves, large waves) of the water texture. Only examples of the "Whale Show" and the "Synchronized Swimming" sequences are shown in this thesis due to copyright issues (cp. Fig. 51).

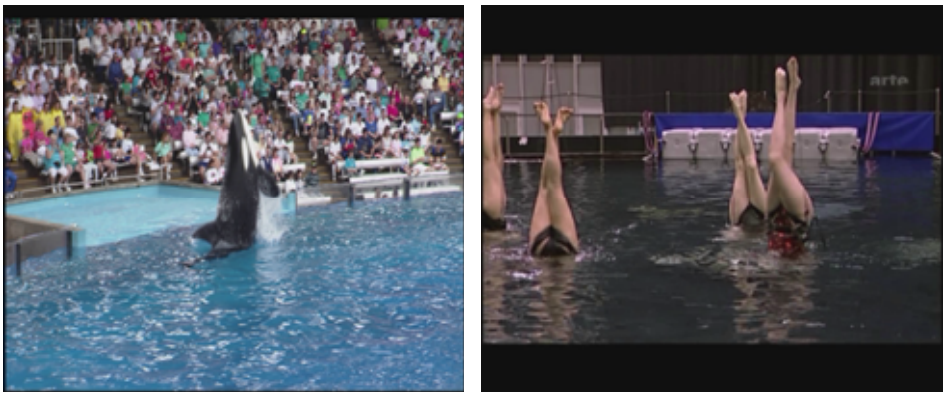


Fig. 51 – Key pictures of test sequences "Whale Show" (left) and "Synchronized Swimming" (right)

Relevant properties of the ground truths are given in Tab. 5. Notice that all the video clips have CIF resolution (352x288), are shown at 15 Hz frame rate and have a length of 6s to 13s.

Video Sequence	Camera motion	Filling strategy	Description
"Talking Man"	Yes	Helical	Talking foreground person (only upper part of the body visible) with sea in the background
"Whale Show"	Yes	Scanline	Orca in oversized swimming pool (foreground) and spectators in the background
"Synchro. Swimming"	No	Helical	Synchronized swimmers in a swimming pool
"Life Belt"	No	Scanline	Life buoy floating on the sea
"Sea"	No	Helical	Person swimming in the sea

Tab. 5 – Properties of test sequences for evaluation of the efficiency of the proposed texture synthesis algorithm for non-rigid textures

The data set is subjectively evaluated using the Double Stimulus Continuous Quality Scale (DSCQS) method (cp. Sec. 6.3.5). That is, test subjects are asked to compare the quality of a synthetic clip with the quality of the corresponding original video sequence. Subjective opinion scores are obtained as a result. Perceptual degradations due to texture synthesis can thus be measured.

5.3.5.2 Configuration of the Texture Synthesizer

As the filling strategy does not have a significant influence on the selected test sequences, the different approaches, i.e. helical and scanline stuffing, are applied arbitrarily. The spatio-temporal boundary condition is sized at least 16 pixels spatially and eight pixels temporally. The patch size is set to 32x32x16 (height x width x temporal depth). Feathering is done in a 3x3x3 neighborhood. Finally, the ref burst length is set to 20 pictures, while the synth burst length lies between 40 and 160 pictures, i.e. 2.7s to 10.7s, depending on the video sequence.

5.3.5.3 Constrained Texture Synthesis without Camera Motion

The video sequences without camera motion (cp. Tab. 5) are evaluated in the first part of the experiments to measure the perceptibility of the distortions inserted through texture synthesis. Ten test subjects have been asked to evaluate the three given test sequences. The experimental results are given in Fig. 52.

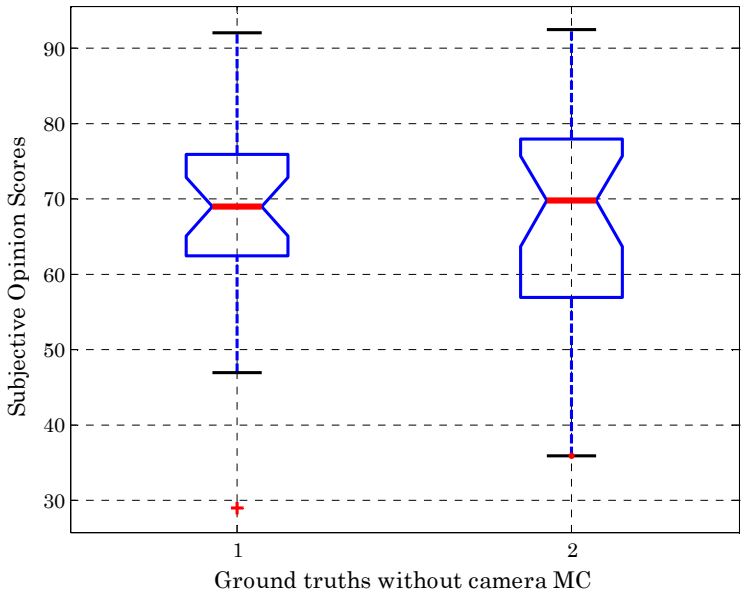


Fig. 52 – Boxplots of the opinion scores obtained using the DSCQS method for non-rigid texture synthesis without temporal alignment. Synthetic sequences (1), reference sequences (2).

It can be seen that no statistically relevant difference between original and synthetic video clips can be observed by the test subjects. Some test subjects have assigned some reference video clips worse opinion scores than corresponding

synthetic versions (cp. lower whiskers of both boxplots in Fig. 52). This shows that the synthesis operation for the constrained framework that is developed in the present thesis can be applied to natural video sequences with good synthesis results.

5.3.5.4 Constrained Texture Synthesis with Camera Motion

Similar experiments as in the previous section are conducted for the sequences with camera motion. Good results are obtained, when temporal alignment of the pictures to be synthesized is conducted, as can be seen in the boxplots 3 and 4 in Fig. 53. No statistically relevant distinction between the opinion scores for reference and synthetic clips can be made in this case. The scores, however, appear to be worse for video clips with camera motion compared to clips without camera motion (cp. Fig. 52 and Fig. 53, boxplots 3 and 4). This might relate to the interpolation operations that are conducted in the course of the motion compensation process.

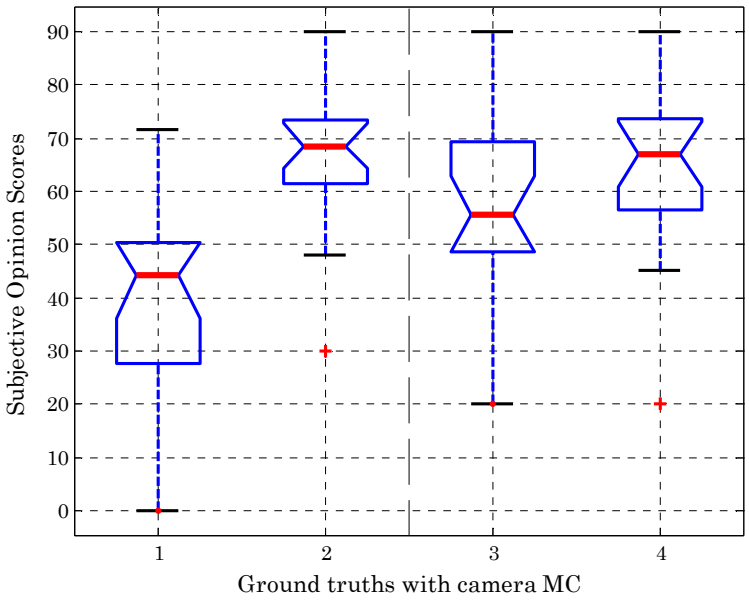


Fig. 53 – Boxplots of the opinion scores obtained using the DSCQS method for non-rigid texture synthesis with and without camera motion compensation. Synthetic sequences without camera motion compensation (1), reference sequences (2), synthetic sequences with camera motion compensation (3), reference sequences (4).

The quality of the synthetic clips decreases rapidly, when no temporal alignment is conducted although camera motion is present in the given video clip. This is

depicted in the boxplots 1 and 2 in Fig. 53. Hence, the temporal alignment feature developed in the present work is important for the perceived quality of a synthetic video sequence. It is interesting to note that the subjective quality scores assigned to the reference video clips (cp. Fig. 53, boxplots 2 and 4) fluctuate depending on the quality of the synthetic clips shown to the test subjects.

5.3.6 Discussion

The texture synthesis algorithm proposed by Kwatra et al. [154] has been successfully extended to meet the requirements of a constrained synthesis environment. Contradictions relating to the old seam optimization, found in [154], could however not be solved. This is due to the fact that no metrical cost function could be found that led to sufficiently irregular spatio-temporal min-cuts. Furthermore, video properties as illumination changes and dominant texture motion directions typically yield subjectively annoying artefacts, when synthesized.

The proposed temporal alignment operation conducted prior to synthesis yields significant improvements in case of a moving camera. However, this is achieved at the cost of the size of the synthesizable texture volume. In fact, synthesis of covered and uncovered regions is not straightforward, and thus avoided, in this thesis. This problem may be tackled in future by conducting the non-rigid texture synthesis recursively.

6 Video Quality Assessment

The texture synthesis modules presented in the previous section generate textures that are perceptually similar to their original counterparts but objectively different. Although the texture synthesis algorithms used in this thesis are very efficient, they can still fail to properly synthesize a given texture class. It must thus be ensured that consistent video quality is reliably generated at the decoder end. This can be achieved by validating the output of the texture synthesis modules via objective video quality estimation measures. Hence, in this chapter, new video quality estimation measures are presented and discussed in detail. Their application to synthetic video textures is thereby emphasized. Experimental results are presented at the end of the chapter to document the efficiency of the proposed quality assessors.

6.1 Motivation and Definitions

A typical image processing chain optionally comprises an acquisition, a processing, a compression, a storage, and a transmission step. Images can be degraded at each step of the chain. Due to the large variety of image processing modules, a number of impairment patterns can be observed in practice.

In Analog Image Processing (AIP), established quality evaluation standards are used to measure the annoyance of corresponding distortions. These standards provide mathematical measures such as the differential gain, the differential phase or waveform distortion that feature a high correlation with the human quality perception [171],[172]. As the performance of an analog image or video processing chain can be assumed to be time-invariant [173], the measurement of its characteristics can be conducted using adequate test signals [174]. Unfortunately, the measures used in AIP are of limited help in the Digital Image Processing (DIP) framework due to the distinct nature of occurring artifacts. For DIP, typical artifacts are blockiness, ringing, motion compensation inaccuracies etc. that are highly correlated with the image content [171],[175],[176]. The behavior of DIP systems is time varying as it depends on operational properties of the digital transmission system. In fact are many decisions taken in a transmission system depending on the input signal. Modules like motion

estimation and adaptive quantization yield dynamic behavior of digital transmission systems due to high signal dependency that is strongly correlated with bit rate or error rate. Hence, deterministic test signals can not be used in the digital transmission framework to reliably gauge the operational behavior of the system [174],[177].

The visibility of DIP artifacts is subject to the spatial, temporal or spatio-temporal properties of the digital input image or video content. This can be explained by inherent properties of the Human Visual System (HVS). A straightforward approach for evaluation of the quality of digital video material thus consists in conducting subjective measurements with human subjects. Albeit ideal for precise image quality assessment, subjective measurements of video quality are costly, time consuming and by no means compatible with real-time constraints. Hence, objective measures are required that can predict the perceptual judgment of human viewers. Major efforts have, in fact, been made to establish standardized quality models given the sustained need for efficient digital video quality assessment tools generated by emerging digital video applications (e.g. digital terrestrial and satellite television, video streaming via the internet, video teleconferencing, video on demand, digital watermarking etc.). The Video Quality Experts Group (VQEG) was created in 1997 and collaborates with ITU-T SG 9 and ITU-R WG6Q [178] (Broadcasting Services - Performance Assessment and Quality Control) to tackle this issue. Typical applications for perceptual video quality measures are dynamic monitoring and adjustment of image quality, optimization algorithms for parameter settings in image processing systems or benchmarking image processing algorithms and systems [179].

Video quality estimation algorithms can be classified based on the required knowledge of the source material. Three categories are typically defined in the literature [180],[179],[181],[182],[183]: Full-reference (FR), reduced reference (RR) and no-reference (NR) approaches.

Full-reference approaches require that original and distorted signal are known in order to determine the corresponding video quality measure [184],[185],[179],

[186],[181],[187],[188],[189],[190],[191],[192]. They are usually impractical in applications where the original signal is not available.

Reduced-reference approaches assume that a compact description of the original signal is given [182],[174],[190]. The compact description typically corresponds to meta data that are transmitted as side information to the node of the DIP system (e.g. digital television networks) where quality evaluations are to be carried out. The same description is extracted from the distorted signal and compared to the reference meta data. Compact side information is required in this framework for bit rate minimization. The difficulty of RR approaches resides in the accurate definition of significant features that capture the underlying properties of the original signal such that a perfect match of the meta data implies perceptual identity between original and processed signal.

No-reference measures do not require any information on the original signal [193],[194],[181],[195],[196],[197]. They can thus be seen as intra methods as they are applied only on the distorted signal. These methods are usually not generic and require a priori knowledge of the type of artifacts that can occur in the given framework. NR measures are, however, desirable in practice as no side information is transmitted here. This is particularly suitable for applications as wireless and IP video services where the throughput is limited and transmission costs are very high.

In the content-based video coding framework presented in this thesis, FR perceptual measures can be used, as the original signal is available at the encoder. Hence new spatio-temporal full-reference video quality measures (VQM) are presented in Sec. 6.4. The VQM is optimized in terms of its efficiency and complexity. In the following section, the state-of-the-art in objective quality assessment of digital images is presented to allow a better understanding of the proposed methods and their underlying assumptions.

6.2 Previous Work

Many full-reference approaches are based on the evaluation of the difference signal between the reference and the processed image signal [183],[182]. Prominent representatives of this class of video quality estimation measures are the mean squared error (MSE) and the Peak Signal-to-Noise Ratio (PSNR). They are very widely used because of their easy manageability with respect to implementation and execution speed. It is however known that these measures poorly correlate with the human visual perception [198],[199],[200],[201]. In fact can two distorted images reveal the same MSE value but feature very different artifacts, where one of the artifact types is much more visible than the other type. Some improvements can be achieved by considering the visibility of differences between original and processed signal. The visibility criterion is thereby defined w.r.t. the HVS properties that can in turn be derived from psychophysical experiments [180],[202],[171].

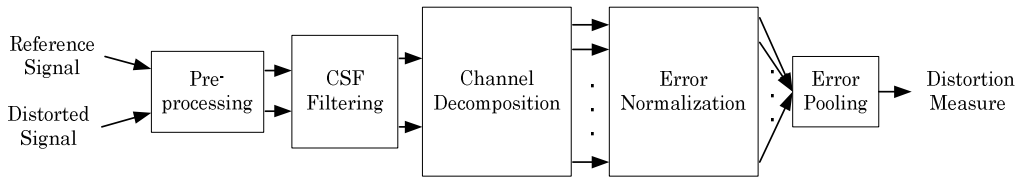


Fig. 54 – Error visibility based perceptual quality evaluation [179]

This type of perceptual video quality assessment tools possesses the same building blocks independently of the approach [179],[189]. Some details may however differ from algorithm to algorithm. The idea here is to model functional properties of early stages of the HVS in order to infer quality perception of given image or video data. For that, error visibility approaches are also called bottom-up methods. It can be seen in Fig. 54 that so-called visibility-based approaches comprise a pre-processing stage, where, for instance, calibration, i.e. spatial and temporal alignment of the distorted image, and color space transformations towards a perceptually homogeneous space are conducted. The next stage is the application of the Contrast Sensitivity Function, also called CSF filtering in the literature [179],[182], to the pre-processed data. The CSF describes human sensitivity to contrast variations. It has been shown that the latter depends on

the color and the spatio-temporal frequencies of the visual stimuli [171]. The third stage is channel decomposition that can be seen as optional depending on the type of video quality model. That is, single channel [189] approaches don't carry out channel decomposition, while multi-channel algorithms do. Multi-channel algorithms a priori match HVS properties better than single-channel approaches [171],[199],[203],[204],[205],[188],[191] as they account for the selectiveness of the HVS for spatio-temporal frequencies and corresponding orientations. The fourth stage is typically error normalization that is conducted channel-wise and where the deviations between original and processed data are determined and normalized w.r.t. masking properties of the HVS. The last stage is called error pooling. This step consists in fusing the error components obtained for each pixel, channel and orientation into one value.

Error visibility approaches are very widespread in the literature [187],[188], [192],[181],[184],[185]. It must however be noticed that the accuracy of such measures fundamentally depends on the acuity with which significant properties of the HVS are described. Nevertheless is it the case that knowledge of the HVS is still in its infancy. Wang et al. [179] for instance point out that the HVS is highly complex and non-linear. Most of the proposed error visibility based models are yet founded on linear or quasilinear operators. Further publications take the same line [206],[207],[192].

In opposition to bottom-up models, top-down approaches have been described in the literature [177],[189],[174],[179],[186]. They typically do not carry out a pixelwise comparison of distorted and reference signal. These approaches compare global disparities between the two signals and thereby usually mimic a selection of assumed HVS functionalities.

6.3 Human Visual System

Discrimination of irrelevant from relevant information is operated in the framework of lossy compression. Relevance should thereby relate to the human perception of visual information. Psychophysical and psychological properties of the human visual system (HVS) should thus be accounted for in the course of the development of lossy compression approaches in particular and image processing algorithms in general.

The eye is the first module of the HV processing chain. The input image is projected onto the retina, that corresponds to a neural tissue at the back of the eye. This tissue is composed of two types of photoreceptors, namely rods and cones. The image on the retina is, in principle, a blurred version of the input signal. The retinal distortion properties are given by the line spread function (LSF) that is measured by using a thin line as the input image [171]. Alternatively, a point can also be used instead of a line, which yields the so-called point spread function (PSF). Mathematical formulations of the LSF and the PSF have been proposed in the literature [171]. Some bottom-up video quality assessment approaches use these functions for pre-processing of the incoming video data [171] (cp. Fig. 54).

The sensitivity of the rods is limited to the brightness of the input image, where they are particularly specialized on low-brightness or scotopic signals. The cones are on the contrary sensitive to higher brightness or photopic signals. The latter photoreceptors are also responsible for color perception. They dominate at the fovea centralis, while the rods are concentrated at the periphery of the retina. Hence, rods and cones are not uniformly distributed across the retina.

The transmission of the input image from the eye to the brain is carried out by neurons and corresponding nerve fibers. The neurons can be seen as local operators as they consider a neighborhood of photoreceptors to generate an output signal. The input signal is conveyed to the visual center of the brain, the visual cortex, where it is processed and analyzed. In fact, detection and recognition tasks are operated in the visual cortex. They encompass edge and line detection as well as discrimination of orientations and frequency bands [4],[171].

6.3.1 Color Perception

In the HVS, color perception is enabled by the cones. There exist three types of these: L, M and S cones. They can be distinguished by their sensitivity to given wavelength ranges. As can be seen in Fig. 55, the L-cones are particularly sensitive to the red primary, the M-cones are particularly sensitive to the green primary, while the S-cones are sensitive to the blue primary (cp. Sec. 4.1.3). The number of cones is significantly lower than the number of rods in the retina (~1:20).

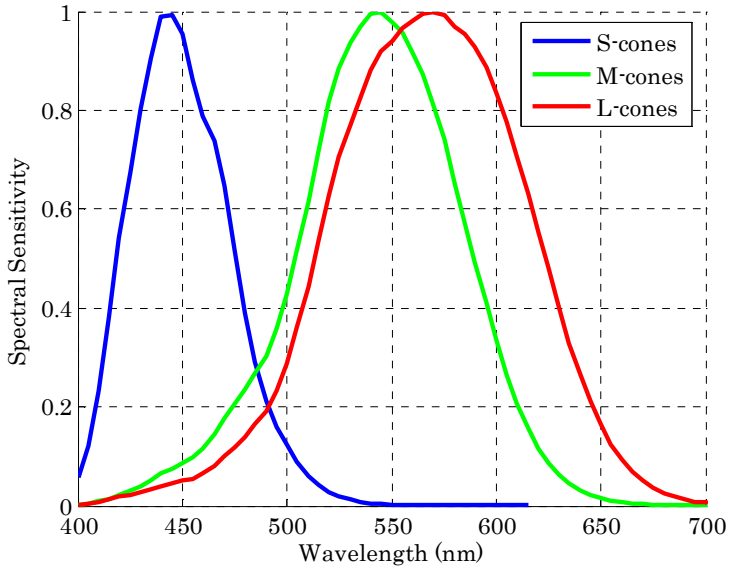


Fig. 55 – Schematic representation of the spectral sensitivity of the cones

6.3.2 Sensitivity Functions

Sensitivity functions describe the HVS's perceptibility of brightness variations. The relative variation of luminance, also called contrast, can not be generically formulated for all stimuli. Hence can several contrast definitions as Weber's or Michelson's be found in the literature [171].

Human contrast sensitivity varies with the adaptation level associated with the local average luminance. Sensitivity to contrast depends on chrominance information as well as spatial and temporal frequencies of the stimuli. Contrast sensitivity functions (CSFs) are typically used to quantify these dependencies. The CSF is the inverse of the contrast threshold, i.e. the minimum contrast necessary for an observer to detect the target. Achromatic contrast sensitivity

typically decreases at low spatio-temporal frequencies, while chromatic sensitivity does not. The full range of colors is perceived only at low frequencies [4],[171].

6.3.3 Multiresolution Analysis

Specialized neurons, called simple cells and located in the primary visual cortex, perform spatially oriented bandpass filtering as well as temporal lowpass and bandpass filtering of the incoming signal. The latter is thus decomposed into spatially orientated and temporal frequency channels. This behavior of the HVS can for instance be incorporated into a perceptual VQA system via carefully designed filter banks [171].

6.3.4 Masking

The masking phenomenon describes interactions between stimuli. Masking is given when a stimulus that is a priori perceptible can not be detected due to the presence of another. Hence, the annoyance of coding artifacts can significantly vary depending on the surrounding textures.

Two types of masking effects can be distinguished, namely spatial and temporal masking. Spatial masking can be seen as an elevation of visibility thresholds of distortions due to spatial discontinuities for instance. Although several spatial masking formulations have been restricted to intra-channel operations, it has been shown in the recent literature that this effect can occur between channels of different orientation or spatial frequency, and also between chrominance and luminance channels [171]. Temporal masking typically occurs in video scenes of high temporal contrast as scene transitions for example. This type of masking effect can be seen as an elevation of visibility thresholds of distortions due to temporal discontinuities in intensity. It has been shown that temporal masking effects do not only set in after a temporal discontinuity but also before its occurrence. This phenomenon is referred to as backward masking in the literature, in opposition to forward masking [171].

It must be pointed out that pooling operations in the brain, i.e. the integration of processing and identification outcomes from multiple channels, are not well understood to date. The different aspects of the HVS presented above correspond to low-level human vision properties. High-level or cognitive properties are more

difficult to capture as they highly vary from individual to individual. It can however be roughly said that human viewers typically focus on regions of high spatial or temporal contrast [171].

6.3.5 Subjective Measurements

The most reliable way to assess subjective quality is conducting subjective viewing tests. For that, a representative group of test subjects must evaluate the quality of the carefully selected video material. Although very time consuming and impractical, subjective tests are the closest we can get to the true perceived quality.

Subjective perception tests are complex to configure because they feature several degrees of freedom that can modify the perceived quality of a given video sequence. These factors must be taken into account to obtain reliable and thus reproducible results. Besides the video content, that is of course of major importance to the viewer, perceived video quality depends on factors like the application, the viewing distance, the video resolution, the brightness, the contrast, the sharpness, the colorfulness, the naturalness etc.

Requirements for subjective viewing tests are defined in the ITU–R Recommendation 500 [208]. Different subjective assessment methods have been defined depending on the given test material. Three of the most frequently used methods are the Double Stimulus Continuous Quality Scale (DSCQS), the Double Stimulus Impairment Scale (DSIS) and the Single Stimulus Continuous Quality Evaluation (SSCQE). A fourth, interesting new method is the Triple Stimulus Continuous Evaluation Scale (TSCES) proposed by Hoffmann [209]. These assessment methods will be presented into some more detail in the following.

6.3.5.1 Double Stimulus Continuous Quality Scale

In this framework, viewers are shown multiple sequence pairs consisting of a reference and a test sequence, which are rather short (typically 10s each). The reference and test sequences are presented twice in an alternating manner. The sequences are separated by a mid-gray level sequence of 3s duration (cp. Fig. 56). Subjects do not have any knowledge of the chronology with which the two sequences are shown.

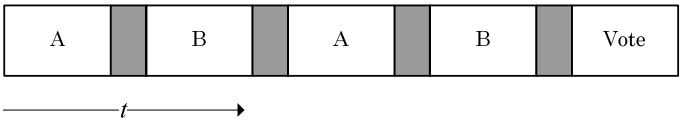


Fig. 56 – Alternating presentation of reference and test signals in a DSCQS trial

The subjects rate each of the two sequences separately on a continuous quality scale ranging from "bad" to "excellent" [208],[4]. Subsequent statistical evaluations are based on the differential rating for each video pair. The latter rating is often calculated from an equivalent numerical scale from 0 to 100 (cp. Fig. 57).

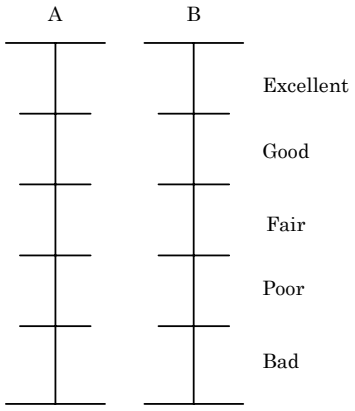


Fig. 57 – Continuous scale for subjective video quality assessment

The DSCQS procedure is the preferred method when the qualities of test and reference sequence are similar.

6.3.5.2 Double Stimulus Impairment Scale

As opposed to the DSCQS, in the framework of the double stimulus impairment scale (DSIS) method, the reference is always shown before the test sequence and neither is repeated [208],[4].

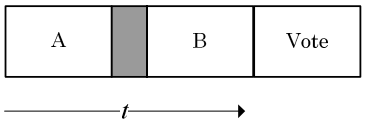


Fig. 58 – Single presentation of reference and test signals in a DSIS trial

Subjects rate the amount of impairment in the test sequence on a discrete five-level scale ranging from "very annoying" to "imperceptible".

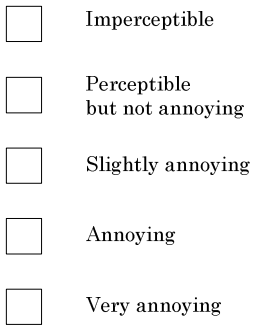


Fig. 59 – Rating scale of the DSIS method

This procedure is preferred for evaluation of clearly visible impairments.

6.3.5.3 Single Stimulus Continuous Quality Evaluation

In this framework, instead of viewing separate short sequence pairs, the test subjects are invited to watch a program of typically 20-30 minutes duration which has been processed by the system under test; the reference is not shown. The subjects must continuously rate the instantaneously perceived quality on the same scale as the DSCQS method (cp. Fig. 57). This procedure is preferred to evaluate the temporal quality variations of digital video compression systems.

6.3.5.4 Triple Stimulus Continuous Evaluation Scale

The triple stimulus method has been proposed by Hoffmann [209] to evaluate HDTV formats and compression systems. Their approach aims to achieve reliable and reproducible results even in scenarios, where different video resolutions are to be evaluated in the same psychophysical test.

In the triple stimulus framework, three video sequences are shown simultaneously on three different displays. The three sequences are synchronized and show the same scene content at the same time. The displays are thereby ordered vertically, i.e. one below the other. The best picture quality is shown on top and constitutes the upper picture quality anchor. The worst picture quality is shown at the bottom and corresponds to the lower picture quality anchor. Finally, the video under test is placed in-between the extreme quality displays. For the upper anchor, uncompressed video is used, while the lower anchor should be selected such that it features similar impairments as the ones expected in the video under test. For consistency evaluation of the subjective ratings, the test sequence set should include the upper and lower anchors. Four repetitions of at least 10s video clips are conducted and the test sequences are shown in randomized order. The continuous quality scale depicted in Fig. 57 is used.

Statistical evaluations are carried out after the subjective evaluations. An in-depth description of the former is given in Sec. 6.5 and Sec. 6.6.

6.4 Proposed Video Quality Assessment Measures

Video quality assessment must be conducted in this thesis to evaluate the consistency of synthetic textures. As explained in Chapter 5, rigid (e.g. sand, flowers, grass etc.) and non-rigid textures (e.g. water, smoke, fire etc.) can be synthesized with corresponding algorithms proposed in this thesis. Although the latter are very efficient, they may fail if, for instance, the segmentation masks are erroneous or the texture to be synthesized violates the underlying assumptions of the texture synthesis methods. In order to guarantee perceptually satisfactory video quality at the decoder end, video impairment estimation for video textures is operated in the analysis-synthesis loop presented in Sec. 3.3.

The artifacts that typically occur in the given framework can be roughly grouped into spatial and temporal categories. Spatial impairments thereby correspond to blocking (tiling) or blurring effects (cp. Fig. 60), while temporal artifacts are typically jerky, unnatural motion.

Fig. 60 – Examples of digital video impairments. Source image (top), blocking or tiling artifact (bottom left), blurring artifact (bottom right) [210].

In this section, video quality measures for both rigid and non-rigid textures are proposed. An in-depth description of the new impairment measures is given in the following.

6.4.1 Overall Method

The overall quality assessment approach for evaluation of synthetic textures is depicted in Fig. 61. The original and distorted sequences as well as a mask series highlighting the synthesized regions are submitted to a Local Spatial Quality Assessment (LSQA) module. The latter corrects the input mask if and where artifacts occur to generate a spatially modified mask (S Mask). The S Mask as well as the input videos are further submitted to the Local Temporal Quality Assessment (LTQA) module.

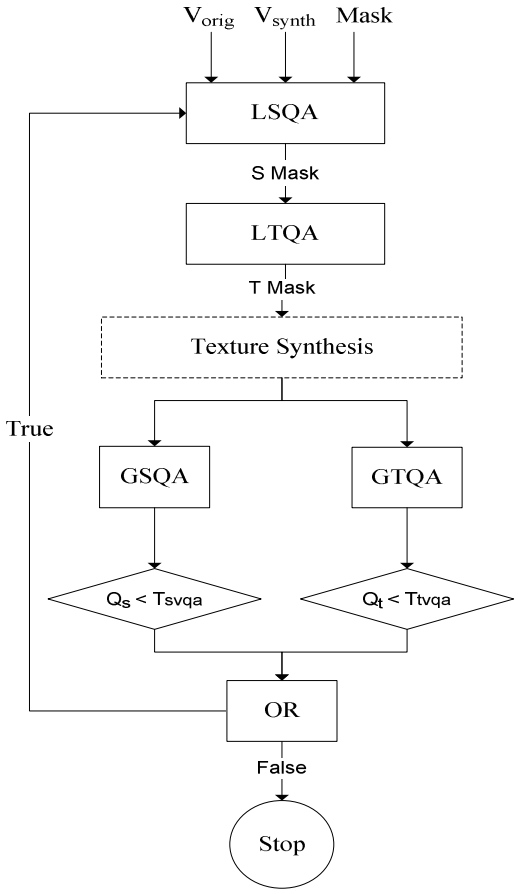


Fig. 61 –Video quality assessment method for synthetic textures

LTQA modifies the S Mask where applicable to generate the T Mask that is used for optional texture synthesis. The global measures are then determined, i.e. the temporal and the spatial measures. The obtained values are now compared to corresponding thresholds, T_{tvqa} and T_{svqa}. Is at least one of the former smaller than the required quality threshold, then the whole quality evaluation is to be repeated. Are on the contrary both quality thresholds met, so is the quality evaluation process stopped and the mask assumed to be of satisfactory accuracy for texture synthesis. Note that the connection between the input videos and the LTQA module is omitted in Fig. 61 for legibility reasons.

6.4.2 Approach by Ong et al.

In this thesis, the VQM defined by Ong et al. [184],[185] is revisited with the aims of simplification, significant performance improvement and adaptation to the given content-based video coding framework. It is, in fact, believed that a much simpler measure can be implemented based on their approach with important gains compared to [184],[185] and other state-of-the-art VQMs [178].

The Video Quality Measure (VQM) proposed by Ong et al. [184],[185] is a top-down approach that integrates some salient properties of the HVS. That is, the hypothesized overall HVS behavior is simulated. This type of approach is preferred in this thesis in order to avoid detailed formulation of assumptions on sparsely understood functional properties of early vision stages, as no comprehensive HVS model exists to date. Published models are typically founded on simplified assumptions that ignore the high complexity and non-linearity of the HVS [179],[171],[206],[207],[192].

The mathematical formulation of the VQM by Ong et al. [184],[185], for a single picture, is given as

$$Q_s(t) = \frac{e^{\frac{\gamma}{1+\delta(t)}}}{e^{\gamma}} \quad . \tag{97}$$

The term γ can be freely selected and steers the interval of $\delta(t)$ for which the contrast is enhanced or reduced (cp. Fig. 62). The denominator in (97) depicts a normalization factor. The variable $\delta(t)$ represents a differential term that assesses the distance between a given reference and a corresponding distorted signal. The definition of $\delta(t)$ is given in (98), where $E_o(t)$ and $E_d(t)$ correspond to mean absolute spatial gradients in the original and the distorted signal respectively. $\delta(t)$ typically features values that lie in the interval [0,1]. However, the latter can not be guaranteed as slight overshoots may occur depending on the data. Given the domain of $\delta(t)$, the domain of $Q_s(t)$ can be given as $\left[e^{-\frac{\gamma}{2}},1\right]$. Notice that $Q_s(t)$ must be maximized to ensure good video quality at the decoder end.

$$\delta(t) = \frac{\left| \mathrm{E_o}(t) - \mathrm{E_d}(t) \right|}{\mathrm{E_o}(t)} \tag{98}$$

$\mathrm{E_o}(t)$ and $\mathrm{E_d}(t)$ are defined as

$$\mathrm{E_o}(t) = \frac{1}{MN} \sum_{\xi=1}^{M} \sum_{\eta=1}^{N} \left| \mathrm{o}_{0^\circ}(\xi,\eta,t) \right| \mathrm{m}(\xi,\eta,t) + \frac{1}{MN} \sum_{\xi=1}^{M} \sum_{\eta=1}^{N} \left| \mathrm{o}_{90^\circ}(\xi,\eta,t) \right| \mathrm{m}(\xi,\eta,t) \tag{99}$$

and

$$\mathrm{E_d}(t) = \frac{1}{MN} \sum_{\xi=1}^{M} \sum_{\eta=1}^{N} \left| \mathrm{d}_{0^\circ}(\xi,\eta,t) \right| \mathrm{m}(\xi,\eta,t) + \frac{1}{MN} \sum_{\xi=1}^{M} \sum_{\eta=1}^{N} \left| \mathrm{d}_{90^\circ}(\xi,\eta,t) \right| \mathrm{m}(\xi,\eta,t) \tag{100}$$

where

$$\mathrm{d}_{0^\circ}(\xi,\eta,t) = \left(\mathrm{d}(\xi,\eta,t) * \mathrm{f}_{0^\circ}(\xi,\eta) \right) \mathbf{C}(\xi,\eta,t) \wedge \mathrm{o}_{0^\circ}(\xi,\eta,t) = \left(\mathrm{o}(\xi,\eta,t) * \mathrm{f}_{0^\circ}(\xi,\eta) \right) \mathbf{C}(\xi,\eta,t) \tag{101}$$

$$\mathrm{d}_{90^\circ}(\xi,\eta,t) = \left(\mathrm{d}(\xi,\eta,t) * \mathrm{f}_{90^\circ}(\xi,\eta) \right) \mathbf{C}(\xi,\eta,t) \wedge \mathrm{o}_{90^\circ}(\xi,\eta,t) = \left(\mathrm{o}(\xi,\eta,t) * \mathrm{f}_{90^\circ}(\xi,\eta) \right) \mathbf{C}(\xi,\eta,t) \tag{102}$$

and

$$\mathbf{C}(\xi,\eta,t) = \frac{\left| \mathrm{o}_{90^\circ}(\xi,\eta,t) \right| + \left| \mathrm{o}_{0^\circ}(\xi,\eta,t) \right| + \left| \mathrm{o}_{45^\circ}(\xi,\eta,t) \right| + \left| \mathrm{o}_{135^\circ}(\xi,\eta,t) \right|}{4} \,. \tag{103}$$

In (99) and (100), $\mathrm{o}(\xi,\eta,t)$ represents the original, while $\mathrm{d}(\xi,\eta,t)$ constitutes the distorted signal. $\mathrm{o}_\beta(\xi,\eta,t)$ and $\mathrm{d}_\beta(\xi,\eta,t)$ $\left(\beta \in \{0^\circ, 45^\circ, 90^\circ, 135^\circ\} \right)$represent the highpass filtered original and distorted signals respectively. $\mathrm{m}(\xi,\eta,t)$ is a binary mask that defines the regions-of-interest. Its components are set to one if such a region is given and to zero otherwise. (M,N) represent the width and the height of the video signal. The highpass filtering operation is defined in (101) and (102), where $\mathrm{f}_\beta(\xi,\eta)$ is a linear, anisotropic gradient filter of orientation β used for edge detection. "*" thereby represents the convolution operation. Contour matrix $\mathbf{C}(\xi,\eta,t)$ is determined from the original signal as defined in (103) and operates object contour enhancement in the original and distorted edge masks. The assumption thereby is that objects and particularly their boundaries are of salient relevance for subjective quality perception [211],[212]. Hence, locations featuring spatial discontinuities are assigned a high weight, while other locations

are assigned low weights. It must, however, be noted that this impairment detector fails if a noisy source signal is given, where object contours can not be properly identified. Given (98)-(103), the VQM defined in (97) applies to single pictures.

The global quality score, Q_s, defined for an entire video sequence, is given as follows

$$Q_s = \frac{1}{L_v}\sum_{p=0}^{L_v-1} Q_s(pD_v/L_v) \text{ with } t = pD_v/L_v \tag{104}$$

where L_v corresponds to the total number of pictures in the given video sequence, while D_v is its total duration in seconds. The quotient D_v/L_v gives the time interval between two pictures.

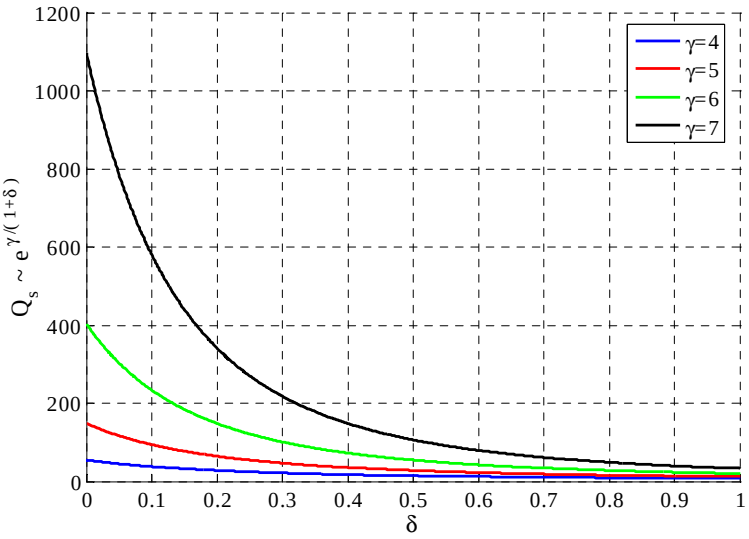


Fig. 62 – Properties of the block fidelity measure

This VQM formulation corresponds to the generalized block fidelity measure proposed by Ong et al. in [184],[185]. A simple gradient filter is used in their proposal. Their VQM is furthermore constrained to the block-based video coding framework. Hence, they formulate (99 and (100) as

$$E_o(t) = \frac{1}{M(\frac{N}{4}-1)}\sum_{\xi=1}^{M}\sum_{\eta=1}^{\frac{N}{4}-1}\left|o_{0^\circ}(\xi,4\eta,t)\right| + \frac{1}{(\frac{M}{4}-1)N}\sum_{\xi=1}^{\frac{M}{4}-1}\sum_{\eta=1}^{N}\left|o_{90^\circ}(4\xi,\eta,t)\right| \tag{105}$$

and

$$E_d(t) = \frac{1}{M(\frac{N}{4}-1)} \sum_{\xi=1}^{M} \sum_{\eta=1}^{\frac{N}{4}-1} \left| d_{0^\circ}(\xi, 4\eta, t) \right| + \frac{1}{(\frac{M}{4}-1)N} \sum_{\xi=1}^{\frac{M}{4}-1} \sum_{\eta=1}^{N} \left| d_{90^\circ}(4\xi, \eta, t) \right| \quad (106)$$

respectively, where the mask $m(\xi, \eta, t)$ is neglected as the whole picture is considered as the region of interest here. 4x4 macroblock transitions are considered in this formulation in order to detect possible blocking effects. Ong et al. [184],[185] sample the highpass masks in parallel to the gradient direction, i.e. horizontal edge masks are sampled vertically, while vertical edge masks are sampled horizontally. Their approach comprises two further measures that aim to capture properties of the HVS (e.g. masking). The former are pooled with the block fidelity measure in a multiplicative manner. One of the additional measures, the so-called distortion invisibility term, however, is very complex as it incorporates color, spatial and temporal masking. The formulation of the different masking properties features more than 20 degrees of freedom, which appears to be hardly manageable.

6.4.3 Global Quality Measures

6.4.3.1 Spatial Artifacts

The first modification proposed in this thesis applies to the gradient filter used in (101) and (102). A sophisticated filter like Sobel is preferred to achieve robustness against spurious edges. The selected edge detector is linear and anisotropic and approximates the gradient of local luminance. Sobel has smoothing abilities, which makes it relatively insensitive to noise [4].

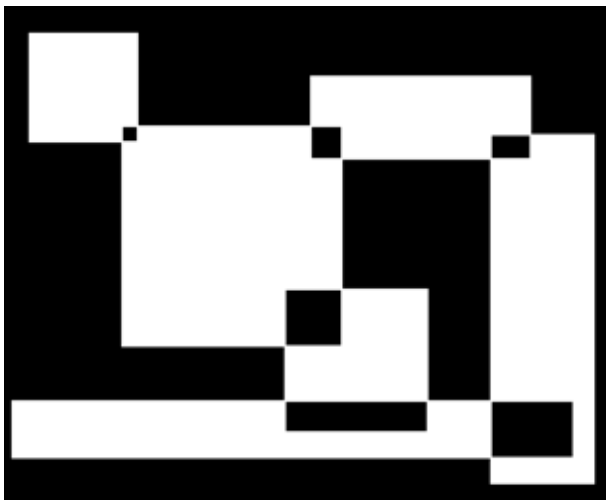


Fig. 63 – Example of a toy input image to be analyzed

Disturbing artifacts like blocking can be detected by the block fidelity measure as already explained above. As the contrast of the pictures plays an important role in quality perception [179],[182], a way must be found to purposely integrate this impairment detection ability into the block fidelity measure. For that, it is noted that the block fidelity measure formulated by Ong et al. [184],[185] inherently detects blurring artifacts, when high frequency areas, that are particularly affected by lowpass effects, are sampled. If $E_d(t)$ is larger than $E_o(t)$ in (98), this indicates that the distorted signal has gained some additional edges compared to the original picture. This may be related to tiling effects in the distorted signal. On the other hand, if $E_d(t)$ is smaller than $E_o(t)$, it can be assumed that a loss of contrast has occurred between original and distorted signals.

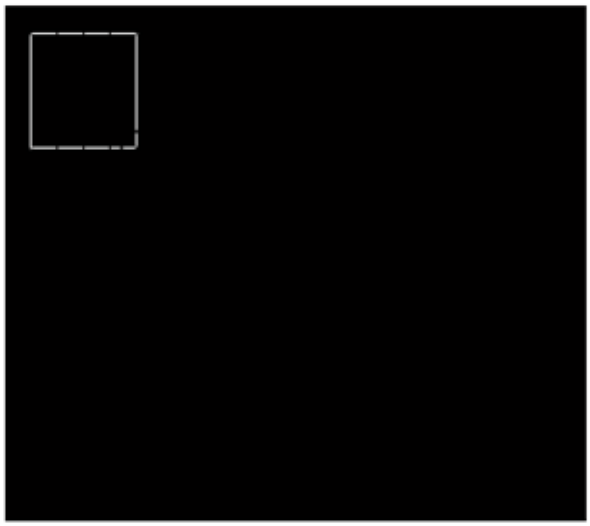


Fig. 64 – Inaccurate sampling of structural information by Ong et al. [184],[185]

The contrast loss detection property of the block fidelity measure is however limited by the fact that structural features like object boundaries are not properly sampled in [184],[185]. Let Fig. 63 exemplarily depict an artificial image that is to be analyzed using the block fidelity measure. Fig. 64 then depicts the response obtained by applying the sampling approach defined in (105) and (106). Although the sampled images are the basis of the block fidelity measure, major structural information is ignored by this measure if the structures don't match the macroblock grid. However, this will assumingly be the case in most natural images. Hence, in this thesis, sampling of the edge masks orthogonally to the gradient's direction is proposed.

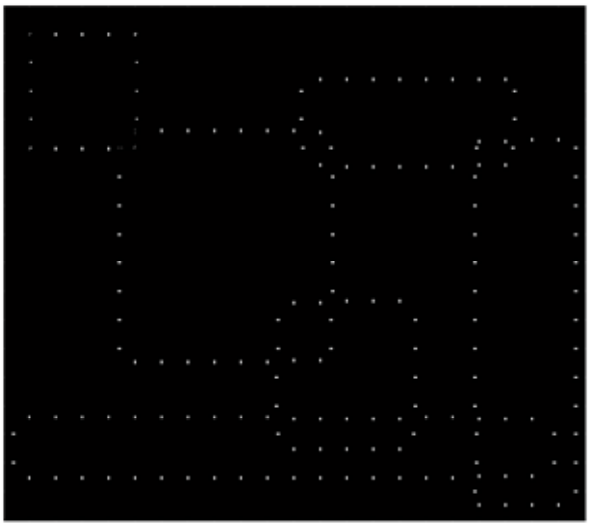


Fig. 65 – Accurate sampling of structural information by the proposed VQM

Object boundaries (white samples) that are particularly important for subjective quality perception [211],[212] are better preserved by the new approach as can be seen in Fig. 65. The blocking detection feature points (gray samples in Fig. 66) constitute a subset of the overall set of feature points selected for quality assessment in this scenario. The proposed feature point selection can be formalized as follows

$$\mathrm{E_o}(t)=\frac{1}{(\frac{M}{\kappa}-1)N}\sum_{\xi=1}^{\frac{M}{\kappa}-1}\sum_{\eta=1}^{N}\left|\mathrm{o}_{0^\circ}(\kappa\xi,\eta,t)\right|\mathrm{m}(\xi,\eta,t)+\frac{1}{M(\frac{N}{\kappa}-1)}\sum_{\xi=1}^{M}\sum_{\eta=1}^{\frac{N}{\kappa}-1}\left|\mathrm{o}_{90^\circ}(\xi,\kappa\eta,t)\right|\mathrm{m}(\xi,\eta,t) \quad (107)$$

and

$$\mathrm{E_d}(t)=\frac{1}{(\frac{M}{\kappa}-1)N}\sum_{\xi=1}^{\frac{M}{\kappa}-1}\sum_{\eta=1}^{N}\left|\mathrm{d}_{0^\circ}(\kappa\xi,\eta,t)\right|\mathrm{m}(\xi,\eta,t)+\frac{1}{M(\frac{N}{\kappa}-1)}\sum_{\xi=1}^{M}\sum_{\eta=1}^{\frac{N}{\kappa}-1}\left|\mathrm{d}_{90^\circ}(\xi,\kappa\eta,t)\right|\mathrm{m}(\xi,\eta,t). \quad (108)$$

where a $\kappa \mathrm{x} \kappa$ macroblock size is assumed.

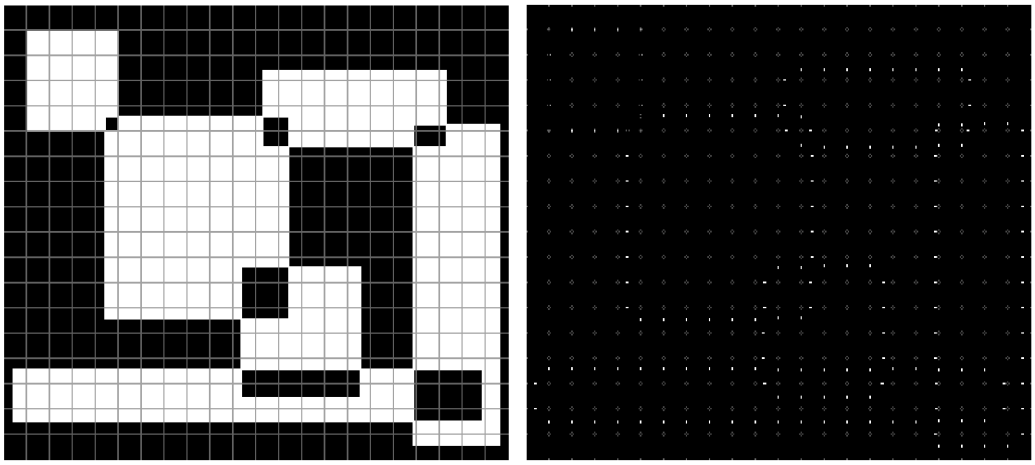


Fig. 66 – Accurate sampling of structural information and macroblock transitions by the proposed VQM

Note that due to the purposeful inclusion of boundary samples into the set of feature points used for quality assessment, not only can tiling effects be detected by the new measure but also distortions affecting object boundaries. Hence, the initial block fidelity measure has been generalized to achieve a simple, generic impairment detection tool.

6.4.3.2 Temporal Artifacts

The proposed temporal VQM consists in evaluating the motion properties of the synthetic textures by matching them with the motion properties of their original counterparts. This is carried out by detecting possible motion inconsistencies based on temporal gradients. For that, ξ-t and η-t slices are defined to assess complex motion. As can be seen in Fig. 67, these slices are determined by considering one of the spatial coordinates to be fixed, i.e. the ξ component in the η-t case and the η component in the ξ-t constellation. In any case do these slices have a temporal extension as potential temporal distortions are to be detected.

The slice definition can be formalized as follows

$$s'_{\eta,o}(\xi,\eta,t,\beta) = s_{\eta,o}(\xi,\eta,t) * f_{\beta}(\eta,t) \quad \wedge \quad s'_{\xi,o}(\xi,\eta,t,\beta) = s_{\xi,o}(\xi,\eta,t) * f_{\beta}(\xi,t) \tag{109}$$

and

$$s'_{\eta,d}(\xi,\eta,t,\beta) = s_{\eta,d}(\xi,\eta,t) * f_{\beta}(\eta,t) \quad \wedge \quad s'_{\xi,d}(\xi,\eta,t,\beta) = s_{\xi,d}(\xi,\eta,t) * f_{\beta}(\xi,t) \tag{110}$$

where $s_{\eta,o}(\xi,\eta,t)$ and $s_{\eta,d}(\xi,\eta,t)$ represent η-t slices in the original and the distorted video signals respectively. $s_{\xi,o}(\xi,\eta,t)$ and $s_{\xi,d}(\xi,\eta,t)$ can be interpreted correspondingly. $s'_{\eta,o}(\xi,\eta,t,\beta)$, $s'_{\xi,o}(\xi,\eta,t,\beta)$, $s'_{\eta,d}(\xi,\eta,t,\beta)$ and $s'_{\xi,d}(\xi,\eta,t,\beta)$ correspond to the highpass filtered slices. Fig. 68 depicts merged η-t and ξ-t slices of the original "canoe" video sequence for the two main diagonal orientations $(\beta \in \{45°, 135°\})$. The latter features a large global translational motion in ξ direction and local motion of the canoeist. It can be seen that salient locations that feature higher motion activity are brighter than such with less motion.

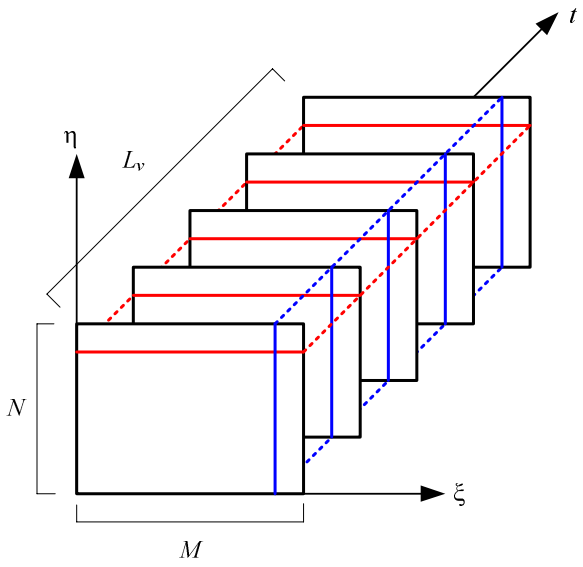


Fig. 67 – ξ-t (red) and η-t (blue) slices in a video sequence

The directional global motion activity at a given picture transition Δt, can be given as

$$\mathrm{GMA}^{o}_{\xi}(\Delta t,\beta) = \frac{1}{K}\sum_{\xi=1}^{M}\sum_{\eta=1}^{N}\left|s'_{\xi,o}(\xi,\eta,\Delta t,\beta)\right| m(\xi,\eta,\Delta t),$$

$$\mathrm{GMA}^{o}_{\eta}(\Delta t,\beta) = \frac{1}{K}\sum_{\xi=1}^{M}\sum_{\eta=1}^{N}\left|s'_{\eta,o}(\xi,\eta,\Delta t,\beta)\right| m(\xi,\eta,\Delta t),$$

$$\mathrm{GMA}_{\xi}^{\mathrm{d}}(\Delta t,\beta)=\frac{1}{K}\sum_{\xi=1}^{M}\sum_{\eta=1}^{N}\left|\mathrm{s}'_{\xi,\mathrm{d}}(\xi,\eta,\Delta t,\beta)\right|\mathrm{m}(\xi,\eta,\Delta t),$$

$$\mathrm{GMA}_{\eta}^{\mathrm{d}}(\Delta t,\beta)=\frac{1}{K}\sum_{\xi=1}^{M}\sum_{\eta=1}^{N}\left|\mathrm{s}'_{\eta,\mathrm{d}}(\xi,\eta,\Delta t,\beta)\right|\mathrm{m}(\xi,\eta,\Delta t), \tag{111}$$

with $K \leq MN$ and $\Delta t = t - (t - \frac{1}{f_s})$. $\mathrm{m}(\xi,\eta,\Delta t)$ represents a binary mask that is set at locations of interest and constitutes the union of the considered picture pair. K is the number of samples set in the binary mask and $f_s = L_v / D_v$ is the frame rate. Notice that boundary issues in preceding filtering operations are neglected. The overall global motion information for a given time interval Δt and slice type can now be formulated as

$$\mathrm{GMA}_{\xi}^{\mathrm{o}}(\Delta t)=\frac{1}{N_\beta}\sum_{n=1}^{N_\beta}\mathrm{GMA}_{\xi}^{\mathrm{o}}(\Delta t,\beta_n),$$

$$\mathrm{GMA}_{\eta}^{\mathrm{o}}(\Delta t)=\frac{1}{N_\beta}\sum_{n=1}^{N_\beta}\mathrm{GMA}_{\eta}^{\mathrm{o}}(\Delta t,\beta_n),$$

$$\mathrm{GMA}_{\xi}^{\mathrm{d}}(\Delta t)=\frac{1}{N_\beta}\sum_{n=1}^{N_\beta}\mathrm{GMA}_{\xi}^{\mathrm{d}}(\Delta t,\beta_n),$$

$$\mathrm{GMA}_{\eta}^{\mathrm{d}}(\Delta t)=\frac{1}{N_\beta}\sum_{n=1}^{N_\beta}\mathrm{GMA}_{\eta}^{\mathrm{d}}(\Delta t,\beta_n), \tag{112}$$

where N_β corresponds to the number of considered edge orientations β. $\mathrm{GMA}_{\xi}(\Delta t)$ and $\mathrm{GMA}_{\eta}(\Delta t)$ can be seen as the mean motion activity at the transition between two consecutive pictures, for a given a slice orientation. The temporal quality measure for a given time interval Δt and slice type can now be determined as

$$\mathrm{Q}_t^{\xi}(\Delta t)=\frac{\mathrm{e}^{\frac{\gamma}{1+\delta_{\xi}(\Delta t)}}}{\mathrm{e}^{\gamma}},$$

$$\mathrm{Q}_t^{\eta}(\Delta t)=\frac{\mathrm{e}^{\frac{\gamma}{1+\delta_{\eta}(\Delta t)}}}{\mathrm{e}^{\gamma}}, \tag{113}$$

with

$$\delta_{\xi}(\Delta t) = \frac{\left| \mathrm{GMA}_{\xi}^{\mathrm{o}}(\Delta t) - \mathrm{GMA}_{\xi}^{\mathrm{d}}(\Delta t) \right|}{\mathrm{GMA}_{\xi}^{\mathrm{o}}(\Delta t)},$$

$$\delta_{\eta}(\Delta t) = \frac{\left| \mathrm{GMA}_{\eta}^{\mathrm{o}}(\Delta t) - \mathrm{GMA}_{\eta}^{\mathrm{d}}(\Delta t) \right|}{\mathrm{GMA}_{\eta}^{\mathrm{o}}(\Delta t)}. \tag{114}$$

The slice-independent temporal quality measure can then be obtained as the minimum of $\mathrm{Q}_t^{\xi}(\Delta t)$ and $\mathrm{Q}_t^{\eta}(\Delta t)$ to give

$$\mathrm{Q}_t(\Delta t) = \min\left(\mathrm{Q}_t^{\xi}(\Delta t), \mathrm{Q}_t^{\eta}(\Delta t)\right) \tag{115}$$

which implies that the slice type with the largest distortions is selected for the overall quality measurement. This allows reliable detection of horizontal, vertical or complex erroneous displacements of objects in a scene. Where $\mathrm{Q}_t^{\xi}(\Delta t)$ is a measure for horizontal motion and $\mathrm{Q}_t^{\eta}(\Delta t)$ is the same for vertical displacements. The overall temporal measure for an entire video sequence can finally be obtained as

$$\mathrm{Q}_t = \frac{1}{L_v - 1} \sum_{n=1}^{L_v - 1} \mathrm{Q}_t(nD_v / L_v). \tag{116}$$

Fig. 68 – Temporal filtering example (bottom) of two consecutive pictures (top) of the "canoe" video sequence

The proposed approach allows for small deviations between original and synthetic motion, which is in line with the fundamental assumptions of the content-based video coding framework of this thesis.

6.4.4 Local Quality Measures

For the proposed video coding framework, some adaptations of the global spatio-temporal quality assessment methods, proposed in the previous sections, are required. Detection of local artifacts in large synthetic regions must be enabled to assist the texture analysis module in the reliable identification of detail-irrelevant textures. Corresponding approaches are described in the following.

6.4.4.1 Spatial Artifacts

Local spatial artifact detection is conducted by applying the global spatial measure (97) to every pixel location (ξ,η,t) in a given region-of-interest. This can be formalized as

$$Q_s(\xi,\eta,t) = \frac{e^{\frac{\gamma}{1+\delta(\xi,\eta,t)}}}{e^{\gamma}} \tag{117}$$

with

$$\delta(\xi,\eta,t) = \frac{\left|E_o(\xi,\eta,t) - E_d(\xi,\eta,t)\right|}{E_o(\xi,\eta,t)} \quad \text{(cf. (98))} \tag{118}$$

and

$$E_o(\xi,\eta,t) = \left|o_{0°}(\xi,\eta,t)\right| m_{90°}(\xi,\eta,t) + \left|o_{90°}(\xi,\eta,t)\right| m_{0°}(\xi,\eta,t) \quad \text{(cf. (107))} \tag{119}$$

and

$$E_d(\xi,\eta,t) = \left|d_{0°}(\xi,\eta,t)\right| m_{90°}(\xi,\eta,t) + \left|d_{90°}(\xi,\eta,t)\right| m_{0°}(\xi,\eta,t) \quad \text{(cf. (108))} \tag{120}$$

where

$$m_{0°}(\xi,\eta,t) = m(\xi,\eta,t) * f_{0°}(\xi,\eta) \quad \wedge \quad m_{90°}(\xi,\eta,t) = m(\xi,\eta,t) * f_{90°}(\xi,\eta) \ . \tag{121}$$

$m_{0°}(\xi,\eta,t)$ and $m_{90°}(\xi,\eta,t)$ represent highpass filtered masks showing the borders of the region-of-interest, where the region-of-interest is given by $m(\xi,\eta,t)$. Hence, synthetic textures are evaluated only at the borders of the region-of-interest. This relates to the fact that spatial impairments due to texture synthesis typically occur at transitions between original and synthetic textures in

the shape of spurious edges. Note that the target borders can be enlarged by some pixels towards to synthetic texture.

Once all the $Q_s(\xi,\eta,t)$ in the transition area have been computed, suspect locations are determined. The latter are defined as VQM values (117) lower than a given threshold. Suspect pixels are depicted in Fig. 69 as black square pixels within (4x4) blocks numbered from 1 to 12. It can be seen that some blocks have no suspect samples (e.g. blocks 1, 3, 4), while others feature just a few of them (e.g. blocks 2, 9, 12). Block 8 on the contrary hosts an important cluster of suspect samples. Is such a cluster given, then all blocks hosting at least one sample of it are labeled potentially erroneous. Considering blocks 5 and 6 for instance, both would be marked potentially erroneous due to the important concentration of suspect samples in their vicinity (cp. Fig. 69). The critical size of a suspect cluster to be seen as potentially erroneous constitutes a degree of freedom of the local spatial VQM. However has a size of at least 12 pixels shown to yield good results indifferently of given video resolution.

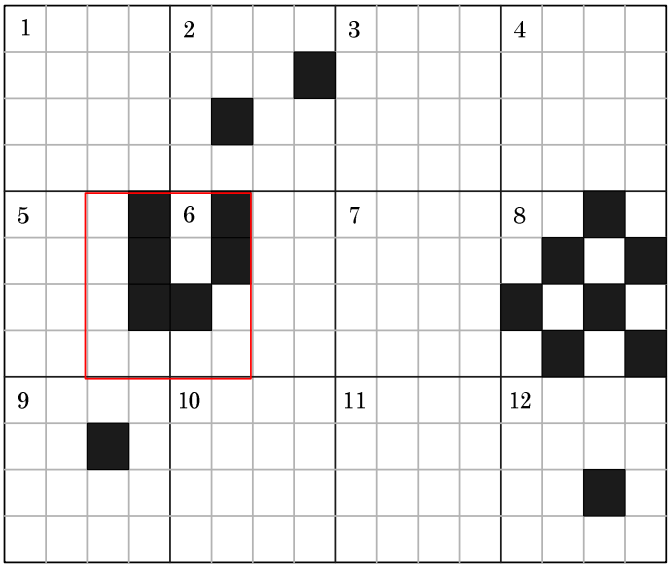


Fig. 69 – Suspect region identification through local spatial quality assessment

For blocks of the region-of-interest that have a low concentration of suspect pixels (e.g. blocks 1, 2, 3, 4, 7, 9, 10, 11, 12 in Fig. 69), a normalized histogram of the $Q_s(\xi,\eta,t)$ values is determined. Notice that this histogram may be populated at very low $Q_s(\xi,\eta,t)$ values, as inconsistent synthetic outcome of limited

concentration is tolerated. Individual, normalized $Q_s(\xi,\eta,t)$ histograms of suspect clusters are further determined (e.g. block 8 in Fig. 69). For that, a patch surrounding each suspect cluster is considered (e.g. red frame within blocks 5 and 6 in Fig. 69). All histograms of suspect locations are matched with the histogram of the validated (low concentration) area using an adequate metric as the l_1 norm for instance. It has shown that the l_1 distance of the histogram centroids is a reliable consistency criterion. Is the distance between the histograms larger than a given threshold (e.g. 0.15), then the given suspect patch is labeled invalid and the blocks sharing common samples with the cluster are marked unsynthesizable. Notice that the histogram's resolution can be varied. It has however shown that no significant gains can be achieved by doing so. Hence, the number of bins is set to 100 per default.

6.4.4.2 Temporal Artifacts

Local temporal artifact detection is conducted by applying the global temporal measure (117)) to single pixel locations (ξ,η,t) in a given region-of-interest. This can be formalized as

$$Q_t^{\xi}(\xi,\eta,\Delta t) = \frac{e^{\frac{\gamma}{1+\delta_{\xi}(\xi,\eta,\Delta t)}}}{e^{\gamma}} \ ,$$

$$Q_t^{\eta}(\xi,\eta,\Delta t) = \frac{e^{\frac{\gamma}{1+\delta_{\eta}(\xi,\eta,\Delta t)}}}{e^{\gamma}} \ , \tag{122}$$

with

$$\delta_{\xi}(\xi,\eta,\Delta t) = \frac{\left| LMA_{\xi}^{o}(\xi,\eta,\Delta t) - LMA_{\xi}^{d}(\xi,\eta,\Delta t) \right|}{LMA_{\xi}^{o}(\xi,\eta,\Delta t)} \ ,$$

$$\delta_{\eta}(\xi,\eta,\Delta t) = \frac{\left| LMA_{\eta}^{o}(\xi,\eta,\Delta t) - LMA_{\eta}^{d}(\xi,\eta,\Delta t) \right|}{LMA_{\eta}^{o}(\xi,\eta,\Delta t)} \ , \tag{123}$$

and

$$LMA_{\xi}^{o}(\xi,\eta,\Delta t) = \frac{1}{N_{\beta}} \sum_{n=1}^{N_{\beta}} \left| s'_{\xi,o}(\xi,\eta,\Delta t,\beta_n) \right| m(\xi,\eta,\Delta t) \ ,$$

$$\mathrm{LMA}_{\eta}^{\mathrm{o}}(\xi,\eta,\Delta t)=\frac{1}{N_{\beta}}\sum_{n=1}^{N_{\beta}}\left|\mathrm{s}'_{\eta,\mathrm{o}}(\xi,\eta,\Delta t,\beta_n)\right|m(\xi,\eta,\Delta t),$$

$$\mathrm{LMA}_{\xi}^{\mathrm{d}}(\xi,\eta,\Delta t)=\frac{1}{N_{\beta}}\sum_{n=1}^{N_{\beta}}\left|\mathrm{s}'_{\xi,\mathrm{d}}(\xi,\eta,\Delta t,\beta_n)\right|m(\xi,\eta,\Delta t),$$

$$\mathrm{LMA}_{\eta}^{\mathrm{d}}(\xi,\eta,\Delta t)=\frac{1}{N_{\beta}}\sum_{n=1}^{N_{\beta}}\left|\mathrm{s}'_{\eta,\mathrm{d}}(\xi,\eta,\Delta t,\beta_n)\right|m(\xi,\eta,\Delta t). \qquad (124)$$

$\mathrm{Q}_t^{\xi}(\xi,\eta,\Delta t)$ and $\mathrm{Q}_t^{\eta}(\xi,\eta,\Delta t)$ are pixel-based temporal quality measures that are determined based on Local Motion Activity (LMA) indicators defined as given in (124). The latter are applied to vertical and horizontal slices, where a set of N_{β} gradient filters is applied to each slice location (ξ,η,t) within the region-of-interest. LMA is given as the linear combination of the outcomes of the gradient filtering operations. The indexes ξ and η of the LMA indicators refer to the considered slice orientation (124).

Once $\mathrm{Q}_t^{\xi}(\xi,\eta,\Delta t)$ and $\mathrm{Q}_t^{\eta}(\xi,\eta,\Delta t)$ have been computed for a given picture pair, suspect locations are determined as local quality measure values lower than a given threshold. The critical size of clusters of such locations to be seen as potentially erroneous constitutes a degree of freedom of the approach. However, a size of at least 12 pixels shows to yield good results. For locations that have been validated by the suspects detection step, a normalized histogram of the $\mathrm{Q}_t^{\xi}(\xi,\eta,\Delta t)$ and $\mathrm{Q}_t^{\eta}(\xi,\eta,\Delta t)$ values is determined. Notice that this histogram may be populated at very low quality values, as poor synthetic textures of limited concentration are tolerated. The histograms of the suspect clusters are further determined individually. The suspect areas are confirmed or rehabilitated as described in the previous section.

Similarly to the local spatial measures, the local temporal impairment predictors $\mathrm{Q}_t^{\xi}(\xi,\eta,\Delta t)$ and $\mathrm{Q}_t^{\eta}(\xi,\eta,\Delta t)$ can be applied to rigid and non-rigid texture synthesizers. It must however be noticed that the mask $m(\xi,\eta,\Delta t)$ is initialized differently in both cases. The fundamental difference between the two synthesizers proposed in the present work resides in the fact that the synthesizer

for rigid textures computes each synthesized picture independently of the others, while the synthesizer for non-rigid textures links subsequent pictures by optimizing their textures simultaneously (cp. Chapter 5). Hence, the mask used for two consecutive pictures corresponds to the union of the single masks for rigid textures. This ensures that all synthesized macroblocks are evaluated by the temporal VQM. The mask for non-rigid texture synthesis is obtained by computing the union of all masks in the considered group of bursts.

6.5 Objective Performance Evaluation of Quality Model

In this section, the verification and validation framework of the proposed video quality assessor is presented. The framework is basically twofold: Prerequisites are first defined for ground truth set data and specific statistical tools are further selected to evaluate the overall performance of the proposed video quality models. The test conditions formulated in this section correspond to the recommendations of VQEG in the so-called Phase II benchmarks [213].

6.5.1 Prerequisites for Ground Truth Data

Video data can feature several distortions after they have been processed. Some of this bias can hamper or even worse impede quality assessment. It must thus be distinguished between impairments that can be removed a posteriori and such that are permanent.

Correctable artifacts can for instance be created, when a source video sequence passes through a codec. The decoded video sequence may then feature a number of scaling and alignment discrepancies compared to the source signal. In order to allow for a fair benchmark of objective quality measurement methods, distorted test data are normalized, where normalization consists in the removal of deterministic disparities between original and distorted signals. Typical correctable artifacts are global temporal frame shift, global horizontal (vertical) spatial image shift, or global chrominance (luminance) gain and offset. The normalized sequences are used for both subjective and objective ratings. Note that the post-processing of the distorted test material is also called calibration.

Artifacts that cannot be corrected through calibration may yield inconsistent responses of the model under test. Hence, data featuring such impairments are

explicitly excluded from the ground truth. Among the permanent artifacts are chrominance differential timing, picture jitter, and spatial re-scaling.

Calibration is not implemented in the present thesis as it does not constitute a primary objective of the present work. Adequate algorithms have been proposed in the literature [213] that could be integrated into the given content-based video coding framework if necessary. Hence, in this thesis, the ground truths selected for performance evaluations of the VQMs do not feature above-mentioned distortions.

6.5.2 Statistical Metrics

In concordance with the VQEG recommendations [213], the proposed model is evaluated based on four major criteria. The first criterion relates to the prediction accuracy, i.e., the ability of the model to reliably predict the subjective video perception. The second criterion refers to the prediction monoticity. This criterion measures the model performance w.r.t. the relative magnitudes of subjective quality ratings. The third criterion, the prediction consistency, refers to the model's robustness against a large variety of impairments. Finally, the fourth criterion relates to classification errors.

6.5.2.1 Prediction Accuracy

The prediction accuracy represents the ability of the quality model to predict the Differential Mean Opinion Score (DMOS) ratings with a reduced error. The MOS is a quantitative measure of the subjective quality of a video sequence as judged by a panel of human observers [213]. The DMOS is in turn a comparative subjective quality measure between a distorted and a reference video sequence.

The criterion recommended by VQEG for prediction accuracy evaluation is Pearson's linear correlation coefficient [213]. The latter metric must be maximized by the model under test and increases given decreasing average errors. Pearson's correlation reflects the degree of linear relationship between two variables. It ranges from +1 to -1. A correlation of +1 or -1 means that there is a perfect positive or negative linear relationship between the considered variables. This relationship is positive when the slope is positive (high scores on the abscissa are associated with high scores on the ordinate) and it is negative when the slope is negative (high scores on the abscissa are associated with low

scores on the ordinate). A correlation of zero means that there is no linear relationship between the two considered variables.

The mathematical formulation of Pearson's correlation coefficient, r_p, can be given as

$$r_p = \frac{\sum_{n=1}^{N_{samp}} (x_n - \mu_x)(y_n - \mu_y)}{N_{samp}\sigma_x\sigma_y} \quad (125)$$

where N_{samp} corresponds to the number of measurements conducted or video sequences in the ground truth. μ_x and μ_y are the mean values of the variables x and y, where these variables correspond to the predicted DMOS, i.e. DMOSp, and the subjective DMOS. σ_x and σ_y are the corresponding standard deviations.

6.5.2.2 Prediction Monoticity

The DMOS predicted by the model under test (DMOSp) should ideally correlate with the corresponding subjective DMOS in terms of sign and magnitude. Spearman's rank correlation coefficient, denoted r_s in the following, is a sensitive monotonicity measure recommended by VQEG [213]. It assesses how well an arbitrary monotonic function could describe the relationship between two variables. Unlike Pearson's coefficient, Spearman's correlation does not require any assumption w.r.t. the functional relationship between the variables (e.g. linear relationship), nor does it require the variables to be measured on interval scales; it can be used for variables measured at the ordinal level. Spearman's coefficient can be seen as a special case of Pearson's correlation as the data are converted to ranks before calculating the former coefficient.

6.5.2.3 Prediction Consistency

Prediction consistency relates to the ability of the quality model under test to provide consistently accurate predictions for a large range of video sequences without excessively failing for a subset of these.

The model's prediction consistency can be measured by the number of outlier sequences. The latter are defined as having an error greater than a given threshold such as one confidence interval. Prediction consistency is given as a

fraction of the total number of sequences. A small outlier fraction means that the model's predictions are relatively consistent.

The prediction consistency criterion recommended by VQEG [213] is the outlier ratio, r_o, defined as follows

$$r_o = \frac{N_{outl}}{N_{samp}} \tag{126}$$

where N_{outl} corresponds to the number of outliers. A video sequence is considered to be an outlier if

$$\left| DMOS_n - DMOSp_n \right| > 2\sigma_{DMOS} \text{ with } n = 0,\ldots,N_{samp} \tag{127}$$

where σ_{DMOS} corresponds to the standard deviation of the subjective DMOS.

It must be noted that, in the Phase II evaluations, VQEG proposes two further metrics besides the ones presented above, namely the root mean square error and the resolving power. They are, however, not further considered in the present thesis because appropriate statements can be made with the four "core" metrics presented above.

6.5.2.4 Classification Errors

In this thesis, classification errors are measured by classifying adequate test data into impaired and non-impaired. Subjective labeling is thereby conducted by operating experiments, where test subjects are asked to make binary decisions given the test data. Classification of the same data is further operated using the VQM under test and mismatches are measured.

6.6 Experimental Results

Experimental evaluations are conducted to validate the proposed spatial and temporal video quality measures. They are first compared to the state-of-the-art in block transform video coding. The latter measures are, however, not conceived for texture synthesis applications. Hence, the proposed measures are further applied to synthetic video sequences in a classification framework, where good and bad synthesis are to be distinguished.

6.6.1 Ground Truth Sets

Unfortunately, the official VQEG (Phase II) test data [213] were not accessible to the author. Hence, four ground truths have been used in the present work to evaluate the video quality measures. Two of the data sets are designed for the block transform coding scenario, where no texture synthesis is operated. The other two sets relate to the analysis-synthesis coding framework. They are compiled to benchmark spatial and temporal artifact detection capabilities of the proposed quality measures.

6.6.1.1 Block Transform Coding

The block transform coding data sets are provided by MPEG [214] and Fraunhofer Heinrich-Hertz-Institut (HHI). The MPEG data set consists of four video sequences and was formerly used to benchmark the performance of MPEG-4 and H.26L anchors [214]. Details on the video sequences selected by MPEG can be found in Tab. 6. The data provided by HHI, also referred to as TV data in the following, correspond to five video clips obtained from several German television channels. The clips feature various contents as news, sports, cartoon, monochrome and color movies that are MPEG-2 coded (cp. Tab. 7). Only examples of MPEG data are shown in this thesis due to copyright issues (cp. Fig. 70).

Fig. 70 – Key pictures of test sequences provided by MPEG. “Container Ship” (top left), “Foreman” (top right), “News” (bottom left), and “Tempete” (bottom right).

Both data sets are subjectively evaluated yielding subjective MOS. The TV data set was evaluated using the DSCQS method, while the MPEG data were evaluated with the DSIS approach (cp. Sec. 6.3.5).

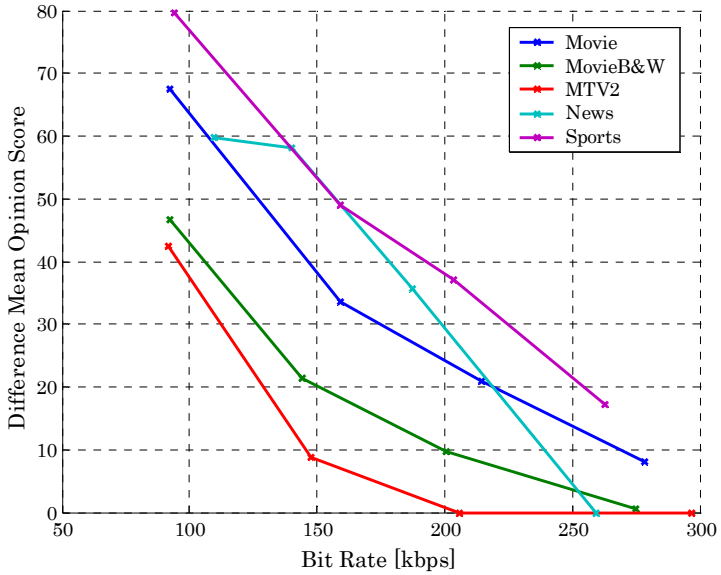


Fig. 71 – Subjective differential MOS vs. bit rate (TV ground truth set)

The TV data are coded with an MPEG-2 codec at five different bit rates per sequence. The highest bit rate is thereby used as reference sequence in the subjective experiments. The MPEG data are coded using an H.26L and an MPEG-4 codec.

Property Designation	Property
Subj. eval. appr.	DSIS
Video codec	MPEG-4 and H.26L
Resolution	QCIF (176x144)
Frame rate	10 Hz and 15 Hz
Duration	10s
Bit rate	32 kbps and 64 kbps
Number of test sequences	4

Tab. 6 – Properties of ground truths provided by MPEG

Subjective video quality variations depending on bit rate fluctuations are depicted in Fig. 71 for the TV data. The same information but for the MPEG data can be found in Tab. 8.

Property Designation	Property
Subj. eval. appr.	DSCQS
Video codec	MPEG-2
Resolution	QVGA (320x240)
Frame rate	12.5 Hz
Duration	10s
Bit rate	Variable
Number of test sequences	5

Tab. 7 – Properties of TV ground truths

6.6.1.2 Texture Synthesis

Two gound truth sets are used for evaluation of the performance of the proposed global spatio-temporal VQMs: Rigid and non-rigid textures. Key pictures of the ground truths are given in Fig. 72, while corresponding information on synthesized textures can be found in Tab. 9. The gound truth sets refer to synthesis results generated by the texture synthesizer for rigid textures and the synthesizer for non-rigid textures respectively.

Sequence	MP4-A	MP4-B	H.26L-A	H.26L-B
Container	3.71	4.12	4.18	4.71
Foreman	1.53	2.12	1.71	3.18
News	2.12	3.06	3.29	4.06
Tempete	3.06	3.53	3.59	4.65

Tab. 8 – Subjective differential MOS (MPEG ground truth set)

Each ground truth set features synthetic textures with imperceptible distortions and such with annoying artifacts. The classification of synthetic results into annoying and imperceptible artifacts was carried out by five students. Impaired test sequences typically feature sporadical artifacts at given time instances.

Video Sequence	Synth. Texture	Synth. Text. Type
"Concrete"	Wall	Rigid
"Flower Garden"	Flowerbed	Rigid
"Husky"	Ground	Rigid
"Canoe"	Water	Non-rigid
"Ducks"	Water	Non-rigid
"Flood"	Water	Non-rigid
"Rain"	Rain	Non-rigid
"Shuttle"	Smoke	Non-rigid
"Synchro. Swimming"	Water	Non-rigid
"Whale Show"	Water	Non-rigid

Tab. 9 – Properties of ground truths for evaluation of efficiency of proposed VQM w.r.t. texture synthesis artifacts

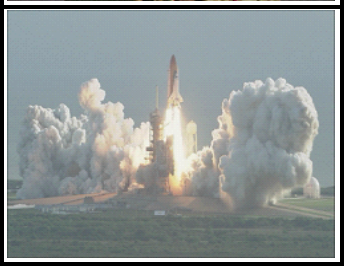
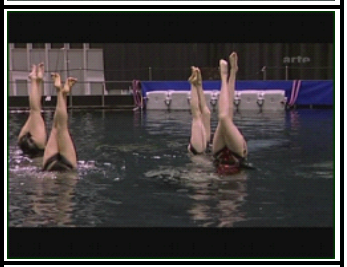
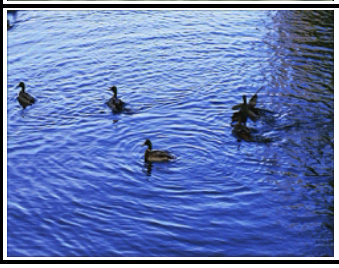
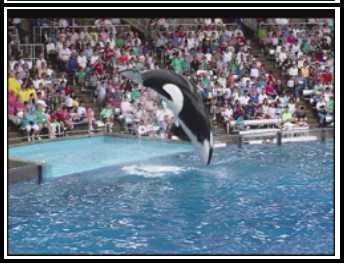
Sequence	Key Frame	Sequence	Key Frame
Concrete		Flood	
Flower Garden		Rain	
Husky		Shuttle	
Canoe		Synchronized Swimming	
Ducks		Whale	

Fig. 72 – Key pictures of test sequences used for evaluation of the efficiency of the proposed VQMs w.r.t. texture synthesis artefacts

6.6.2 Statistical Evaluation Method

The statistical evaluation method of the benchmarked video quality measures is described in this section.

6.6.2.1 Data Normalization

As explained in the previous section, double stimulus subjective evaluation methods have been used for both MPEG and TV ground truths. That is, for each test subject, video sequence and codec setting two opinion scores are available, one for the reference video sequence and one for the coded video (cp. Sec. 6.3.5). The relative difference between the assessed values for source and processed video sequences is used for further evaluations. That is, the absolute difference is normalized by the opinion score of the reference video sequence. This is done to account for individual subjective quality scales of test subjects.

The subjective scores obtained for the MPEG data have been mapped onto the range 1-5 by MPEG for their evaluations. The corresponding MOS thus lie in the same range. The MOS have been mapped onto the range 1-100 in the present thesis in order to achieve comparable results for MPEG and TV data.

A single subjective score is required for each video sequence. This is achieved by computing the median value of the corresponding differential opinion scores achieved from the test subjects. The influence of outlier scores on the differential mean opinion score (DMOS) is limited by this operation.

6.6.2.2 Regression

Given a ground truth, the functional relationship between subjective and predicted MOS can be highly non-linear depending on the video quality assessment model. Comparability of different assessors can be achieved by means of non-linear regression [213]. The proposed VQMs feature an inbuilt logistic function with a degree of freedom γ that steers the linearity of the relationship between subjective and predicted MOS. Reference assessors selected in this thesis are PSNR and the best measure proposed by VQEG [213], i.e. the general model of the National Telecommunications and Information Administration (NTIA). PSNR is defined as

$$PSNR = 10\log_{10}\frac{255^2 MN}{\sum_{\xi=1}^{M}\sum_{\eta=1}^{N}\left[o(\xi,\eta,t) - d(\xi,\eta,t)\right]^2}. \quad (128)$$

As NTIA's general model is nearly linear with respect to subjective data [213] and VQEG's phase II evaluations show that no significant performance improvement of the assessors can be achieved through non-linear regression [213], linear regression will be conducted in the experimental evaluations in the present thesis [77]. Notice that 95% confidence intervals are used for linear regression.

The statistical metrics presented in Sec. 6.5.2 are computed for each regression line determined in the experimental evaluations. In order to ensure reproducibility of the results achieved in the experiments, the cross validation approach [77] is used. The latter approach allows identification of models that perform best on unknown data. This is achieved by evaluating the correlation between the subjective MOS and the predicted MOS (MOSp) with data that have not been used to determine the regression line. Since the ground truth sets are relatively small, a regression line is determined with all sequences available in a given ground truth set except one of the sequences. The sequence left out is used to determine the statistical metrics. The mean statistical metrics are finally determined from the left out sequences. This procedure, where no explicit validation data set can be afforded due to limited ground truths, corresponds to a special case of the cross validation approach called the "leave-one-out" method [77]. Note that the cross validation approach is a statistical tool that can be applied to various other problems that differ from the present one. It should also be noted that the regression line plotted in Fig. 73, for example, is the median regression line obtained from the cross validation iterations.

6.6.3 Block Transform Video Coding

In this section, the proposed video quality measure is evaluated w.r.t. its performance and compared to PSNR and VQEG's best quality assessor proposed by the NTIA. The data used in this section are the TV and the MPEG data that can be used to evaluate assessors based on block transform coded video sequence. As already explained above, these sequences feature artifacts that are relevant for the coding framework of the present thesis. The proposed global spatial VQM is used here. Its configuration is given in Tab. 10. The full image plane is considered for quality assessment.

Parameters	Setting
γ	11
κ	16
$f_{0°}(\xi,\eta)$ (Sobel operator)	$\begin{pmatrix} 0.25 & 0.5 & 0.25 \\ 0 & 0 & 0 \\ -0.25 & -0.5 & -0.25 \end{pmatrix}$
$f_{90°}(\xi,\eta)$ (Sobel operator)	$\begin{pmatrix} -0.25 & 0 & 0.25 \\ -0.5 & 0 & 0.5 \\ -0.25 & 0 & 0.25 \end{pmatrix}$
$f_{45°}(\xi,\eta)$ (Sobel operator)	$\begin{pmatrix} -0.5 & -0.25 & 0 \\ -0.25 & 0 & 0.25 \\ 0 & 0.25 & 0.5 \end{pmatrix}$
$f_{135°}(\xi,\eta)$ (Sobel operator)	$\begin{pmatrix} 0 & 0.25 & 0.5 \\ -0.25 & 0 & 0.25 \\ -0.5 & -0.25 & 0 \end{pmatrix}$

Tab. 10 – Configuration of proposed spatial VQM

Fig. 73 and Fig. 74 depict the regression results obtained for the proposed VQM and both data sets. The 95% confidence intervals (per sequence and quality) and the statistical metrics are also given. It can be seen that high correlation coefficient values are obtained for both Pearson and Spearman. The outlier ratio is zero.

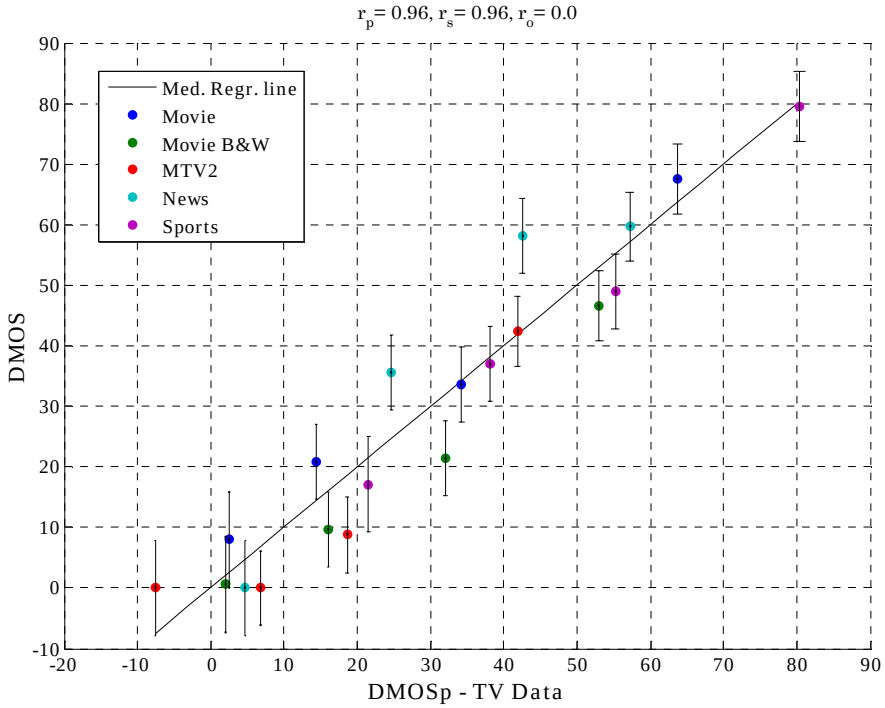


Fig. 73 – Linear regression results for the proposed VQM and the TV ground truth set

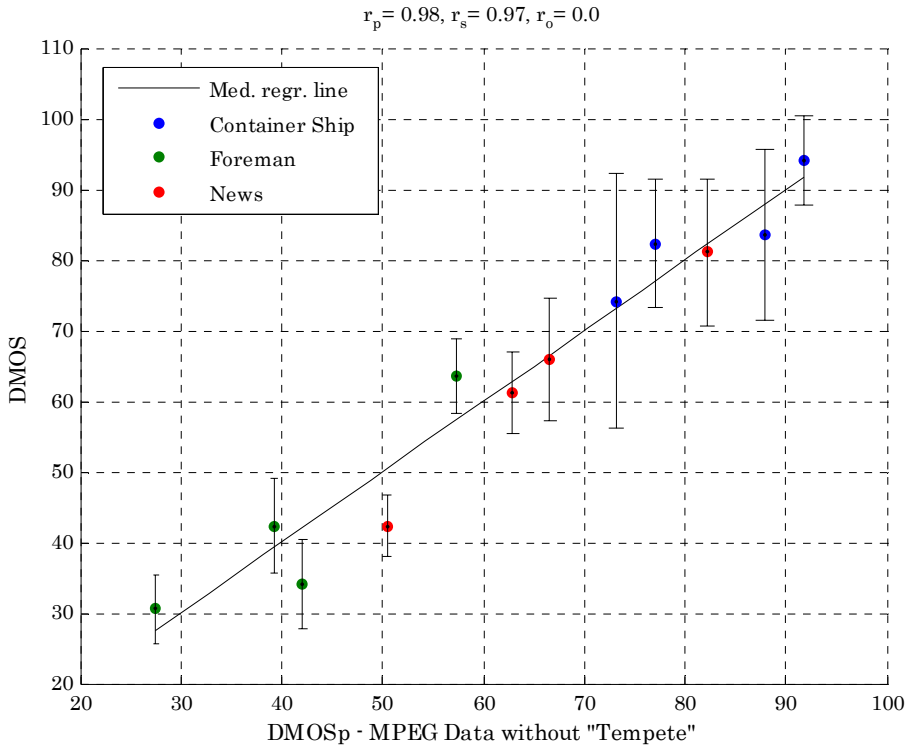


Fig. 74 – Linear regression results for the proposed VQM and the MPEG ground truth set

Fig. 75 and Fig. 76 depict the regression results obtained for PSNR and both data sets. It can be seen that high correlation coefficient values are obtained for the MPEG data. Given the fact that two codec architectures have been benchmarked with these data and PSNR as the distortion criterion, these results appear to be plausible. Pearson's correlation coefficient is identical for PSNR and the proposed VQM. However, the proposed VQM yields better Spearman correlation (+0.02). This implies that the proposed VQM features better rank correlation properties than PSNR on a data set that is adapted to the latter measure. PSNR's performance is significantly degraded in the case of the TV data set. Significantly lower correlation coefficients are obtained. The outlier ratio is zero for both data sets.

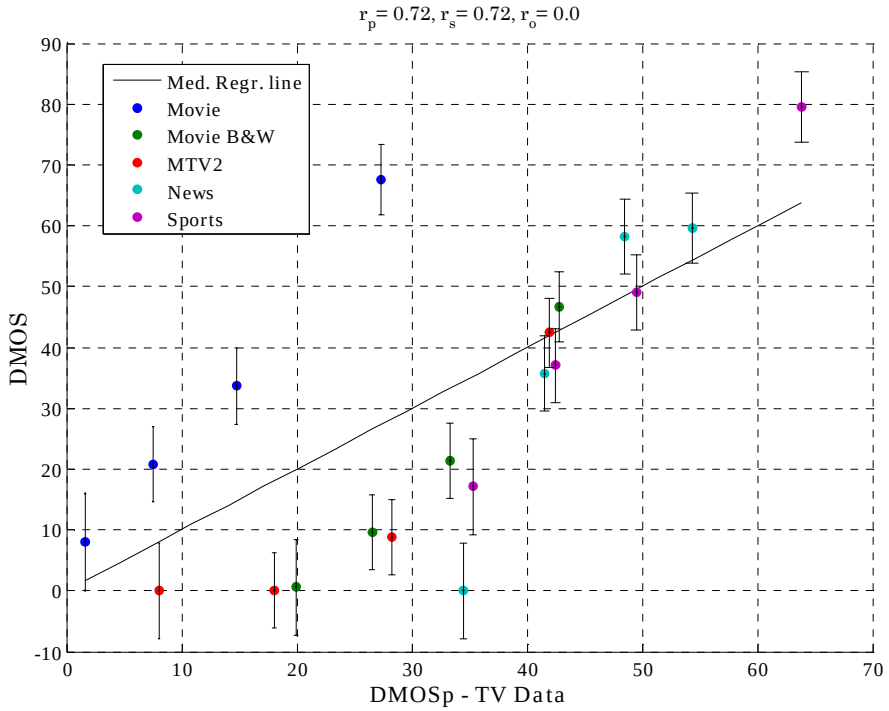


Fig. 75 – Linear regression results for PSNR and the TV ground truth set

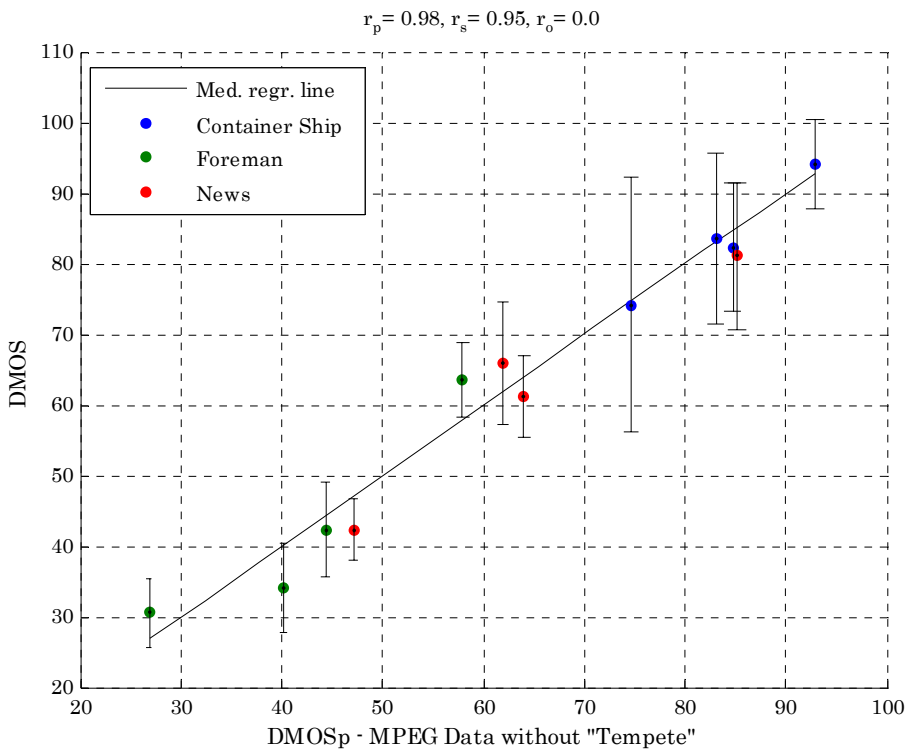


Fig. 76 – Linear regression results for PSNR and the MPEG ground truth set

Fig. 77 and Fig. 78 depict the regression results obtained for NTIA's general model and both data sets. This VQM has been completely implemented in user friendly software that is freely available via NTIA's web site. It can be seen that the general model performs better than PSNR and the proposed VQM on the MPEG data set and w.r.t. Pearson (+0.01). However, Spearman's correlation coefficient is still lower than for the proposed metric (-0.01). The results obtained for the TV data are significantly better than PSNR's but slightly worse than the proposed measure. The outlier ratio is zero for both data sets.

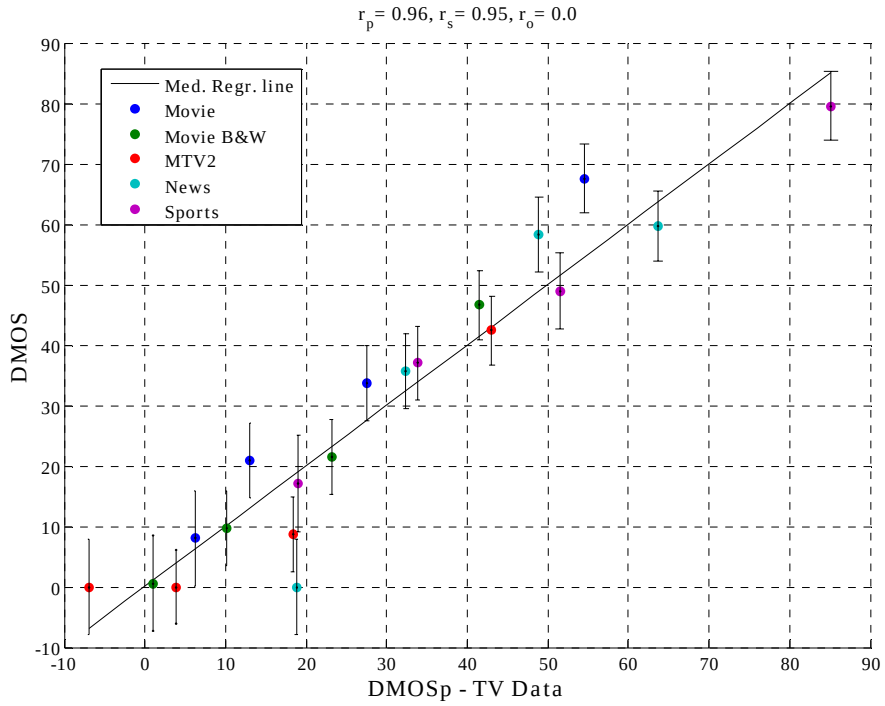


Fig. 77 – Linear regression results for NTIA's general VQM and the TV ground truth set

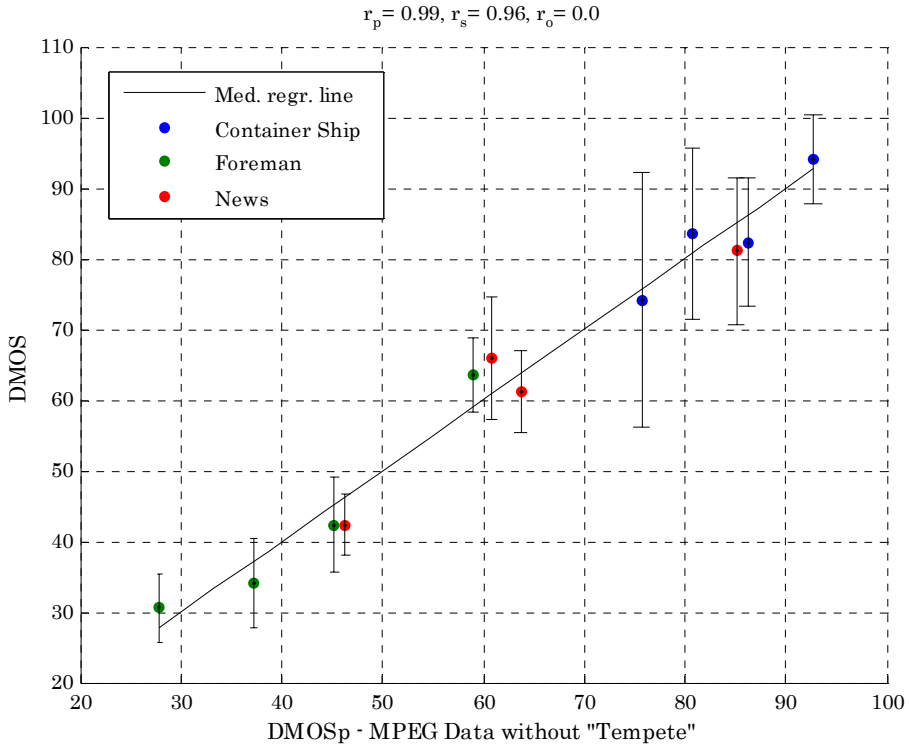


Fig. 78 – Linear regression results for NTIA's general VQM and the MPEG ground truth set

The results presented above are summarized in Fig. 79 and Fig. 80. For the sake of completeness, the results for the measure proposed by Ong et al. [184],[185] are also given. It can be seen that this VQM performs significantly worse than NTIA's and the proposed quality assessor. The outlier ratio is 0.1 for the TV data set and zero for MPEG's ground truth set. It should be noted that Ong et al.'s VQM features more than 20 degrees of freedom. It was not possible to obtain the optimized parameter setting from the authors due to patent issues. Hence default parameters were set in the present thesis. They can be found in Appendix B.

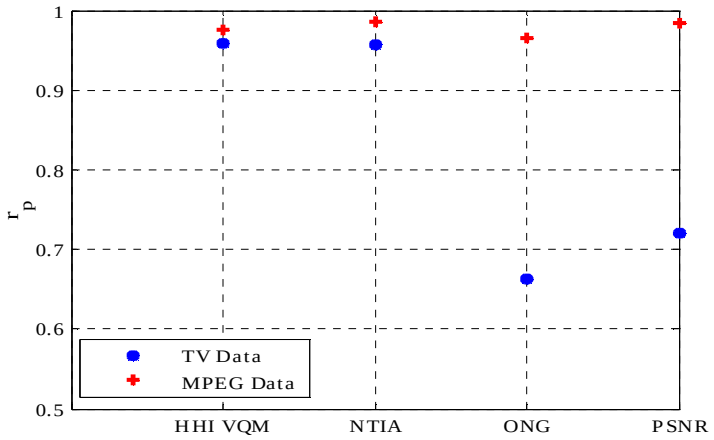


Fig. 79 – Pearson's correlation coefficients achieved for evaluated VQMs

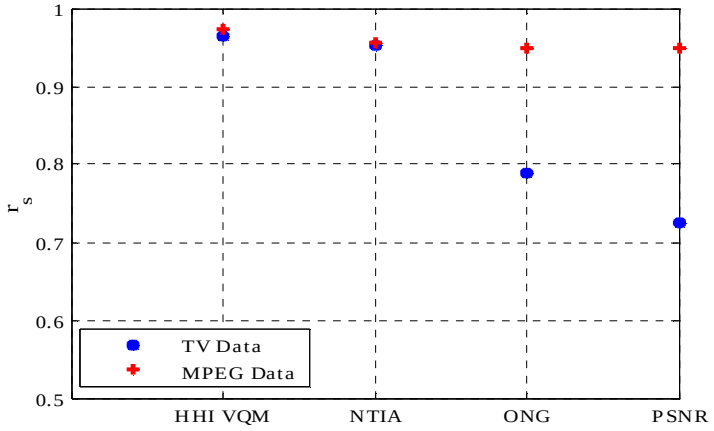


Fig. 80 – Spearman's correlation coefficients achieved for evaluated VQMs

The rank order correlation, i.e. Spearman's correlation coefficient, is particularly important in the present coding framework, as automatic discrimination of bad

synthesis results is required. This can only be achieved if the measured distortions MOSp correlate with MOS in terms of ranking of the distortions. As can be seen in Fig. 80, the proposed VQM yields better r_s results than all other examined VQMs.

The MPEG data set contains a video sequence called "Tempete". A picture of this video is depicted in Fig. 81. Note that the regression line is determined based on the "inlier" sequences here. That is, the sequences for which the hypothesis that the relationship between MOS and MOSp is linear holds (F-test [215]). The "Tempete" sequence was removed from the data set for all evaluations conducted above. This relates to the fact that for all the considered VQMs, the "Tempete" sequence showed a severe outlier behavior such that regression computations led to the conclusion that no linear relationship exists between MOS and MOSp in all cases (F-test [215]).

Fig. 81 – A key picture of the "Tempete" sequence and its corresponding highpass filtered version

A regression result, when "Tempete" is included in the MPEG data set is given in Fig. 82. The failure of all the examined quality metrics can be explained by the extreme characteristics of the video sequence. In the sequence, leaves of relevant resolution are continuously falling in the foreground of the scene. The falling of the leaves is chaotic and hardly predictable by motion estimation algorithms, which yields motion prediction failures. That implies lots of localized errors and low objective quality scores. These leaves are however most probably not the objects of interest in the scene. The former nevertheless contribute important distortions that yield to bad quality predictions although the background of the scene that might be subjectively more relevant may not feature annoying

artifacts. As can be seen in Fig. 81, the leaves yield high responses in $\mathbf{C}(\xi,\eta,t)$ (103) which implies that these particular locations will be assigned high weights in the overall distortion evaluation.

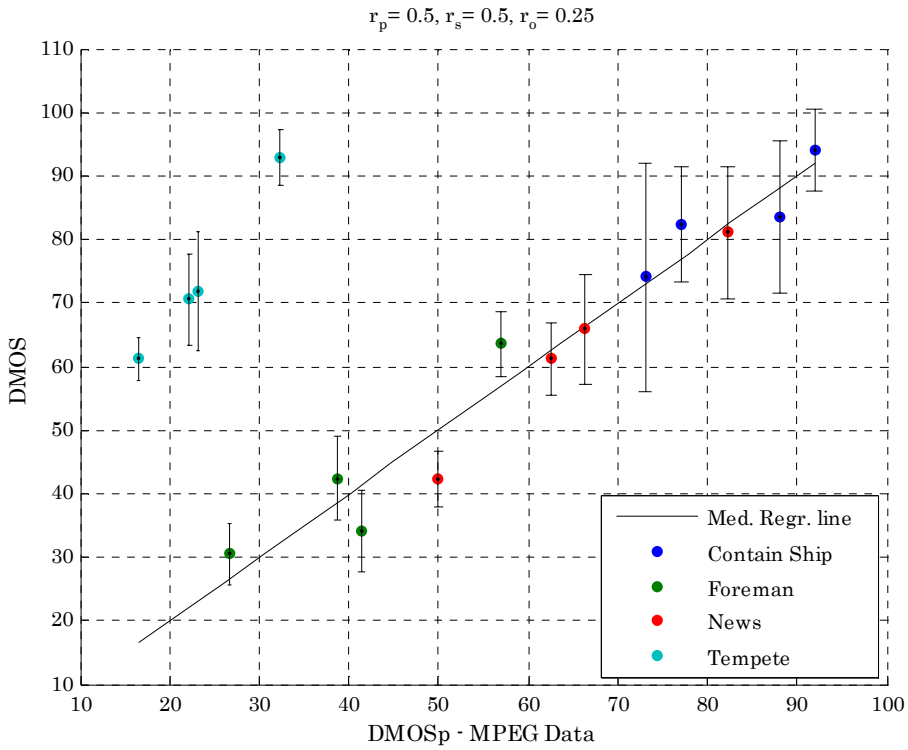


Fig. 82 – Linear regression results for the proposed VQM and the MPEG ground truth set including "Tempete"

6.6.3.1 Complexity

In this section, the complexity of the proposed VQM is compared to the complexities of PSNR and particularly NTIA's quality assessor. This is done by estimating the corresponding complexities per picture. This approach is chosen in the present thesis because no command line version of NTIA's quality measure is freely available, such that no precise complexity evaluations can be conducted using adequate software tools. Meaningful insights can however be obtained through simple approximation as will be shown in the following.

The complexity of PSNR can be determined as

$$Pr^{1}_{PSNR} = 6 + MN;\ Add^{1}_{PSNR} = 2MN \tag{129}$$

where Pr and Add represent the number of products/divisions and the number of additions/subtractions respectively.

The complexity of the proposed VQM can be determined as

$$Pr_{VQM}^{1} \approx \frac{78}{\kappa} MN + 3 = 4.87MN + 3;\ Add_{VQM}^{1} \approx \frac{13}{\kappa} MN + 2 = 1.625MN + 2\ . \tag{130}$$

NTIA's general VQM is given as

$$\begin{aligned} VQM_{NTIA} = & -0.2097\, si\_loss \\ & +0.5969\, hv\_loss \\ & +0.2483\, hv\_gain \\ & +0.0192\, chroma\_spread \\ & -2.3416\, si\_gain \\ & +0.0431\, ct\_ati\_gain \\ & +0.0076\, chroma\_extreme \end{aligned} \tag{131}$$

where the definitions of the terms of the VQM can be found in [174]. They will not be given here as they are not relevant for the subsequent complexity evaluations. The si_loss term comprises a 13x13 linear filter operation. This implies an (under-) estimated complexity of

$$Pr_{NTIA} = 169\, MN;\ Add_{NTIA} = 168\, MN\ . \tag{132}$$

It can be seen in Fig. 83 that both VQMs, i.e. NTIA's and the proposed assessor, are significantly more complex than PSNR in terms of the number of multiplications. As far as additions are concerned, NTIA's VQM is significantly more complex than PSNR, while the proposed VQM features a comparable complexity to PSNR. This can be explained by the fact that only a selection of feature samples are used for quality assessment by the proposed VQM. Notice that the complexity of PSNR has been set to 100% in Fig. 83. The complexity of the two other VQMs is given relatively to PSNR's in the latter figure.

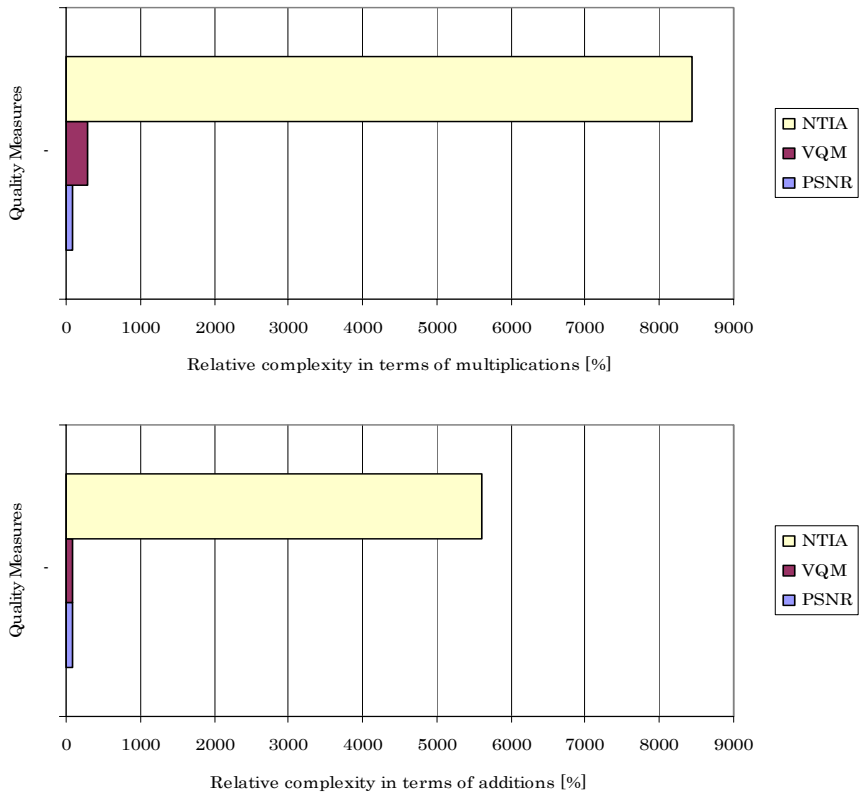


Fig. 83 – Complexity evaluation of relevant video quality assessors. Products/divisions (top), additions/subtractions (bottom).

It also appears that just the filtering operation required in one of the terms of NTIA's VQM is already significantly more complex than the proposed quality assessor. Similarly complex operations are conducted in the other terms of the measure by NTIA. Hence, it can be said that the proposed VQM is significantly less complex than VQEG's best VQM but performs comparably well.

6.6.4 Analysis-Synthesis Framework

In this section, verification and validation of the proposed global spatio-temporal quality metrics for synthetic rigid and non-rigid textures is done. The two ground truths relating to texture synthesis are used in the following evaluations.

6.6.4.1 Rigid Textures

Spatial artifacts occur in the shape of inconsistent transitions between synthetic and natural textures in a video sequence. These unnatural spatial discontinuities can be detected by the proposed global spatial VQMs (cp. Sec. 6.4.3). Its configuration is given in Tab. 10.

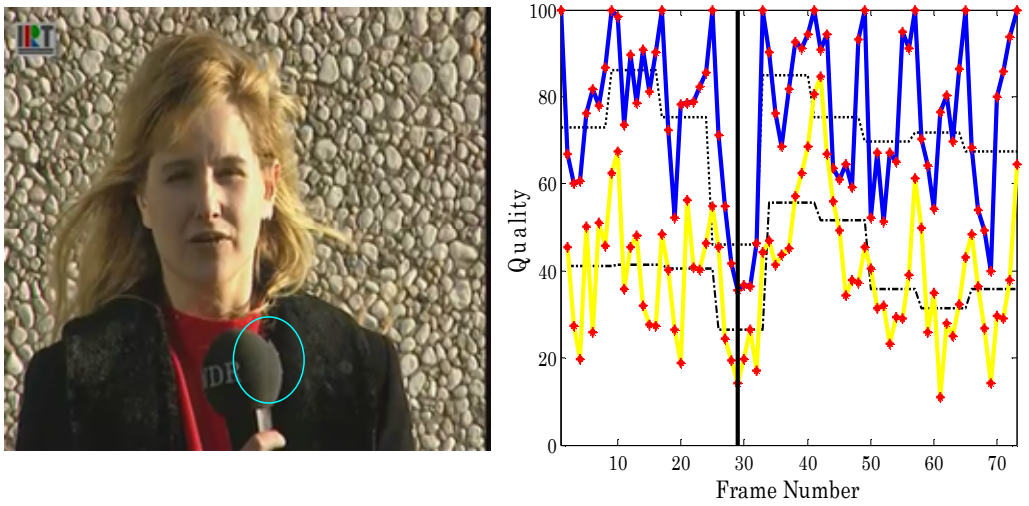


Fig. 84 – Spatial artifact related to rigid texture synthesis in the "Concrete" video sequence

The spatial VQM is applied to the ground truth set for rigid textures (cp. Fig. 72 and Tab. 9). Fig. 84 depicts the results obtained for the "Concrete" sequence. A picture of "Concrete" can be seen on the left hand side of the figure, while the spatial VQM (blue curve) measured over time is given on the right hand side. The yellow curve, representing the global temporal VQM, will be discussed later in this section. Notice that the whole background of the scene is synthetic as well as a part of the foreground object. It can be seen that due to sporadical spatial artifacts in the video, the predicted quality scores fluctuate between high and low values. The picture shown in Fig. 84 is impaired (noticeable impairment marked by a light blue circle). The corresponding low quality score is highlighted by a black vertical line on the right hand side of the same figure. This type of failure of the texture synthesizer for rigid textures can be explained by inaccurate motion estimation (cp. Sec. 5.2). The dotted line corresponds to the mean quality score per GoP.

Another problem related to rigid texture synthesis is that, depending on the type of motion in the video sequence, samples at the picture borders might not be available in the reference pictures (cp. Sec. 5.2). This is a sign that the GoP may be too long given the motion activity in it. If no countermeasures are taken, the

missing samples will yield very annoying artifacts in the shape of white areas. An example of these is highlighted in Fig. 85.

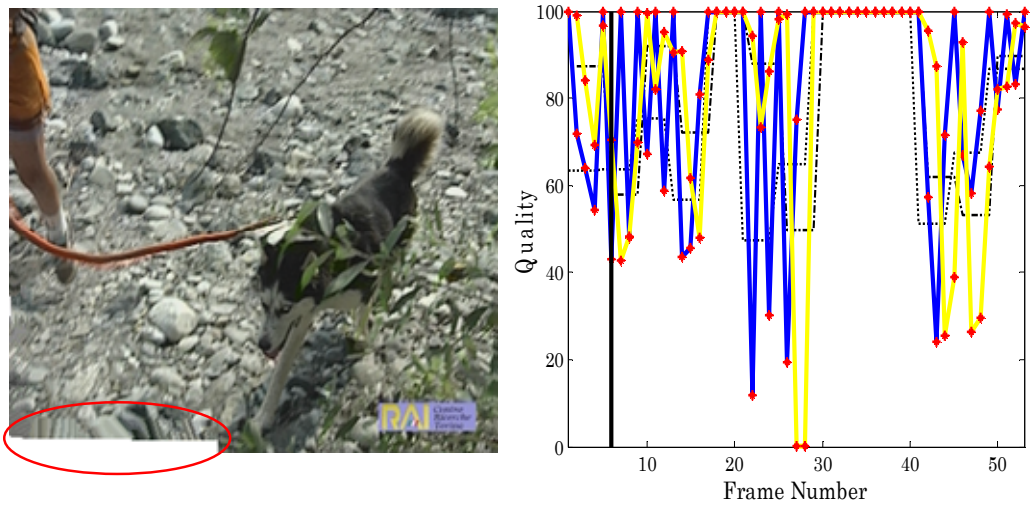


Fig. 85 – Spatial artifact related to covering and uncovering effects in "Husky" video sequence

It can be seen that a low quality score is obtained in such cases. Notice that the whole ground has been synthesized in this sequence. If countermeasures are taken in the shape of intra-synthesis with a Markovian synthesizer (cp. Sec. 5.2), these may still yield unsatisfactory results especially if the region featuring missing samples is too large (cp. Fig. 86).

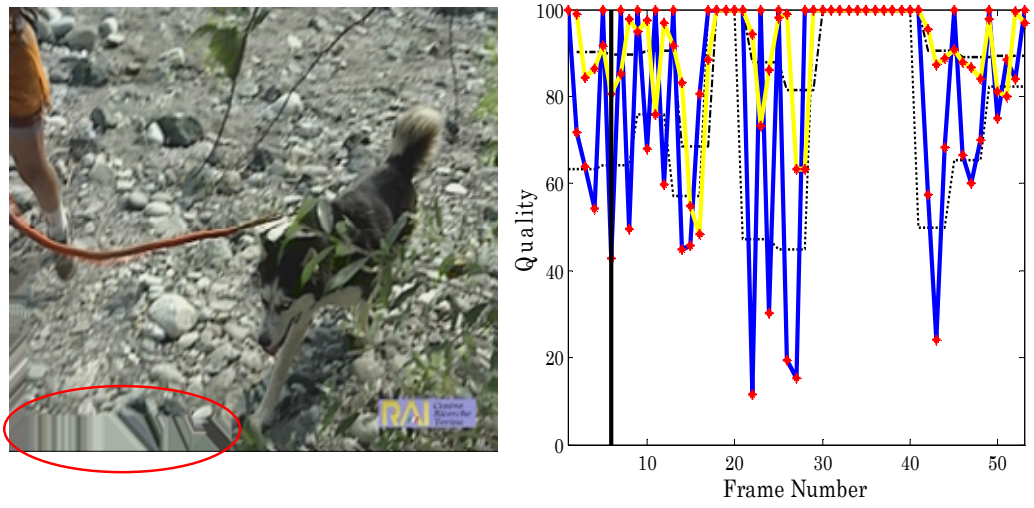


Fig. 86 – Spatial artifact related to covering and uncovering effects and handled with Markovian intra-synthesis post-processing ("Husky" video sequence)

The synthesizer for rigid textures can also yield temporal artifacts in the shape of jerky motion (cp. Sec. 6.4). These impairments can be detected by the proposed global VQM for temporal artifacts. Its configuration is given in Tab. 11. Note that Roberts' cross operators are used here. This is done to reduce the temporal considerations to two consecutive pictures such that the motion information is well captured given the small spatio-temporal neighborhood. The results obtained for the proposed VQM are illustrated considering the "Concrete" sequence as example. Two different temporal transitions of the sequence are considered.

Parameters	Setting
γ	11
$\mathrm{f}_{45°}(\xi,t)/\mathrm{f}_{45°}(\eta,t)$	$\begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}$
$\mathrm{f}_{135°}(\xi,t)/\mathrm{f}_{135°}(\eta,t)$	$\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$

Tab. 11 – Configuration of proposed temporal VQMs

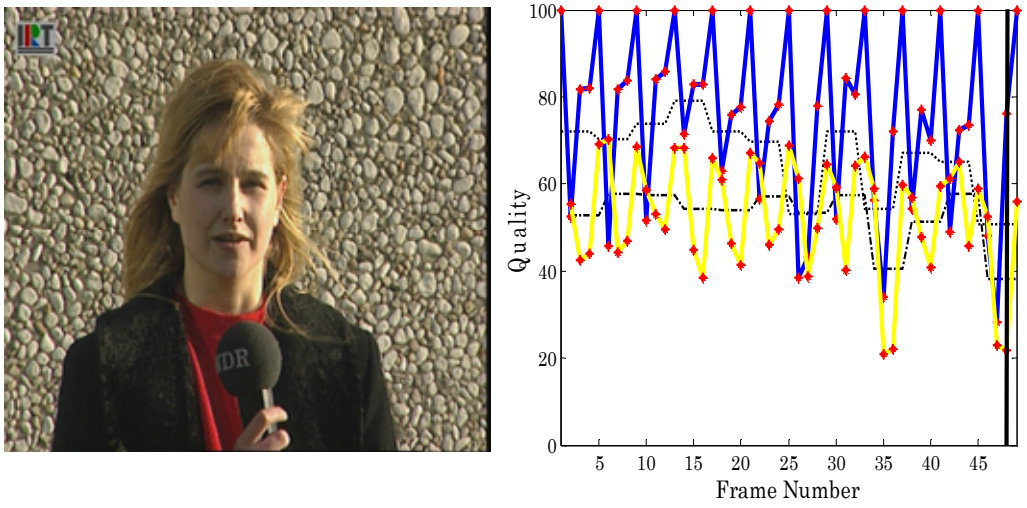


Fig. 87 – Subjectively annoying temporal artifact in the "Concrete" video sequence and corresponding quality score (vertical black line)

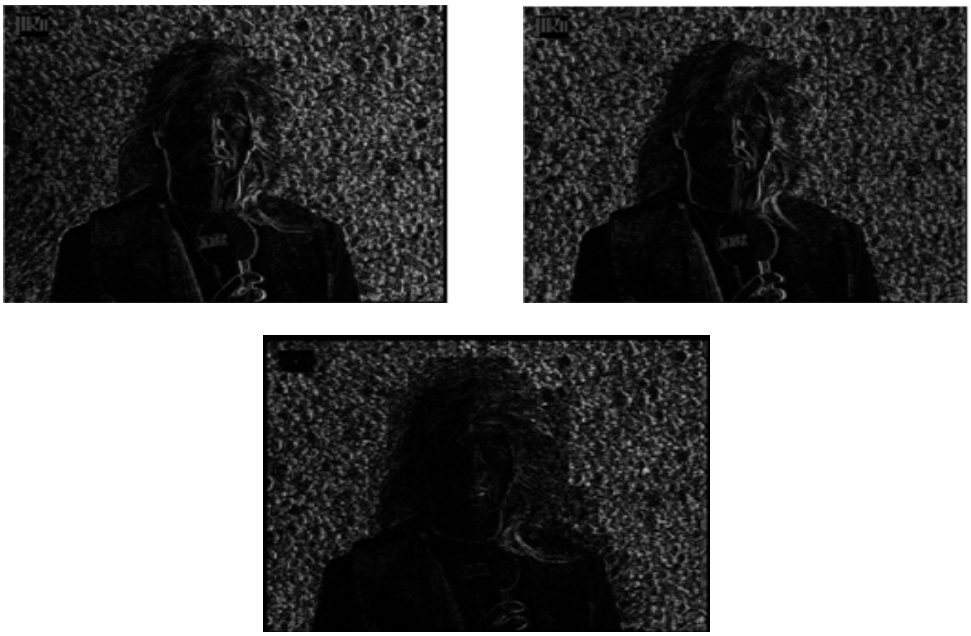


Fig. 88 – Temporal gradients measured for the reference (top left) and the impaired (top right) video sequence and corresponding difference signal (bottom)

It can be seen that the first transition (cp. Fig. 87 and Fig. 88) obviously features a very noticeable impairment as the quality score is very low. On the contrary, the second transition (cp. Fig. 89 and Fig. 90) has been assigned a very high score, which implies that synthesis was successful in that case. The proposed VQM is confirmed by the difference signal between the synthetic and the corresponding reference temporal gradient mask (cp. Fig. 88 and Fig. 90). It can be seen that in the case of the impaired synthesis result, the difference signal features very high amplitudes, while they are almost zero for the successful synthesis. Notice that MSE reflects subjective impairment perception in the selected examples.

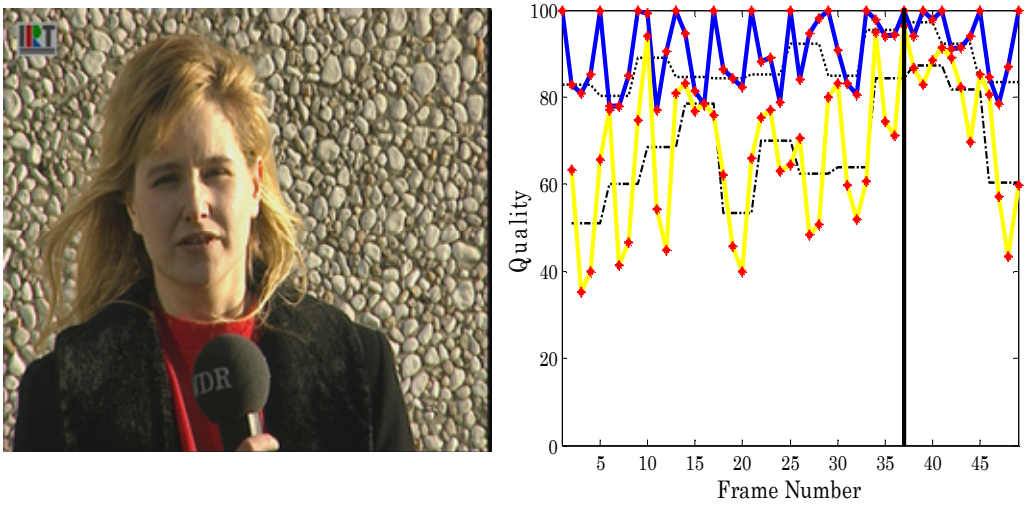


Fig. 89 – Subjectively non-annoying temporal transition in the "Concrete" video sequence and corresponding quality score (vertical black line)

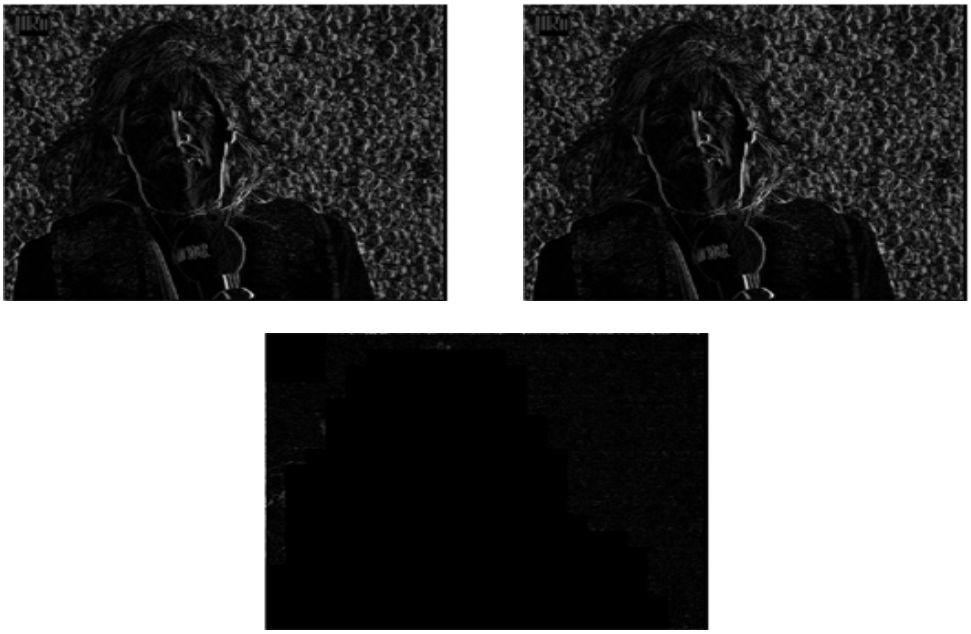


Fig. 90 – Temporal gradients measured for the reference (top left) and the impaired (top right) video sequence and corresponding difference signal (bottom)

6.6.4.2 Non-Rigid Textures

Similarly to synthetic rigid textures, spatial artifacts occur in the shape of inconsistent transitions between synthetic and natural textures in this scenario. Temporal artifacts relate to jerky unnatural motion. The configurations of the spatial and the temporal VQMs are given in Tab. 10 and Tab. 11 respectively. The proposed global spatial and temporal VQMs are applied to the ground truth set for non-rigid textures.

Fig. 91 depicts typical results obtained for non-rigid textures with the "Rain" sequence as an example. The whole background of the scene is synthetic. Obvious spatial artifacts are visible in the given picture. Temporal artifacts are also noticeable in this video sequence. This is reflected by the results obtained with the proposed VQMs in Fig. 91. Notice that the spatial VQM is depicted by the blue curve, while the temporal VQM corresponds to the yellow curve.

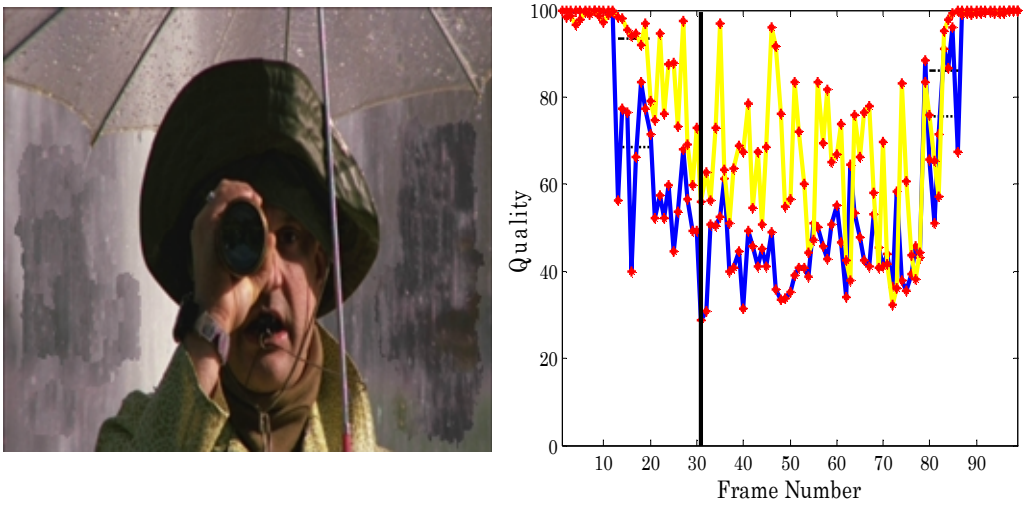


Fig. 91 – Subjectively annoying spatio-temporal artifact in the "Rain" video sequence and corresponding quality scores (vertical black line)

6.6.4.3 Classification

In order to objectively gauge the performance of the proposed global spatial and temporal VQMs, the separability of impaired and good synthesis results is quantitatively evaluated in the following. This is done for both synthesis ground truth sets, i.e. for rigid and non-rigid textures, separately. Group of pictures have been extracted from each of the ground truths and classified into "good" and "bad" synthesis by five students.

The quality scores obtained from the proposed VQMs are given in Fig. 92 for rigid textures. It can be seen that, in this scenario, a clear distinction between good and bad synthesis results can be achieved. This implies that the proposed temporal and spatial VQMs are efficient for corresponding artifact detection given rigid synthetic textures. A discrimination threshold can be chosen in the interval $[72\ 75]$ for spatial artifacts, while it should lie in the interval $[54\ 62]$ for temporal artifacts. This can be verified in Fig. 92 and Fig. 93. Discrimination is slightly worse for temporal artifacts than for the spatial ones but still statistically significant as the boxplot notches are pairwise non-overlapping in Fig. 92 (cp. Sec. 4.4.1).

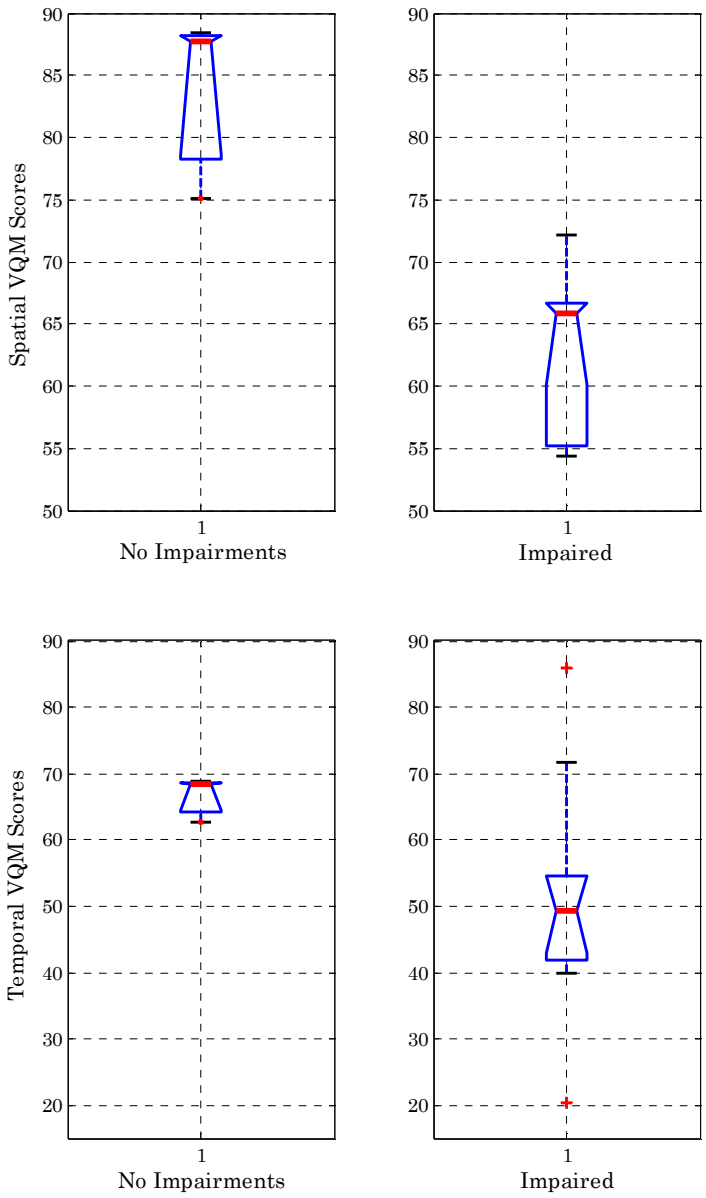


Fig. 92 – Quality scores for synthetic video sequences with and without noticeable impairments (rigid textures)

A synthetic result is considered to be impaired if one of the thresholds is undershot (cp. Fig. 93). The classification results obtained for the test sequences (rigid textures) are given in Tab. 12. The spatial threshold is set to 72, while the temporal threshold is set to 60. It can be seen that a correct classification is obtained for 100% of the data. Each of the artifact assessors detects the class of impairments it is responsible for correctly in all the cases.

Sequence	GoPs	Subj. Score	Global Spatial Measure Qs		Global Temp. Measure Qt		Output
			Overall	Worst GoP	Overall	Worst GoP	
Canoe	7	Bad	11,56	7,993	20,39	5,202	IMPAIRED
Concrete – A	3	Good	88,43	73,64	68,78	50,31	OK
Concrete – B	7	Bad	72,15	46,13	39,92	26,42	IMPAIRED
Concrete – C	3	Good	87,67	80,26	68,41	50,95	OK
Concrete – D	3	Bad	66,62	50,8	52,64	38,24	IMPAIRED
Flower Garden - A	3	Good	75,05	62,42	62,8	48,73	OK
Flower Garden - B	7	Bad	55,26	49,79	47,55	34,72	IMPAIRED
Husky – A	3	Bad	71,48	60,98	50,88	38,67	IMPAIRED
Husky – B ('Bridge' texture)	7	Bad	65,65	62,07	41,87	36,21	IMPAIRED
Husky – C ('Stones' texture)	3	Bad	65,97	59,42	54,51	35,94	IMPAIRED
Husky – D ('Stones' texture)	7	Bad	66,22	60,64	44,02	29,21	IMPAIRED
Husky – E (White blocks)	3	Bad	57,79	47,31	71,62	49,85	IMPAIRED
Husky – F (Markovian intra synthesis)	3	Bad	54,41	44,89	85,99	68,64	IMPAIRED

Tab. 12 – Classification results for rigid textures

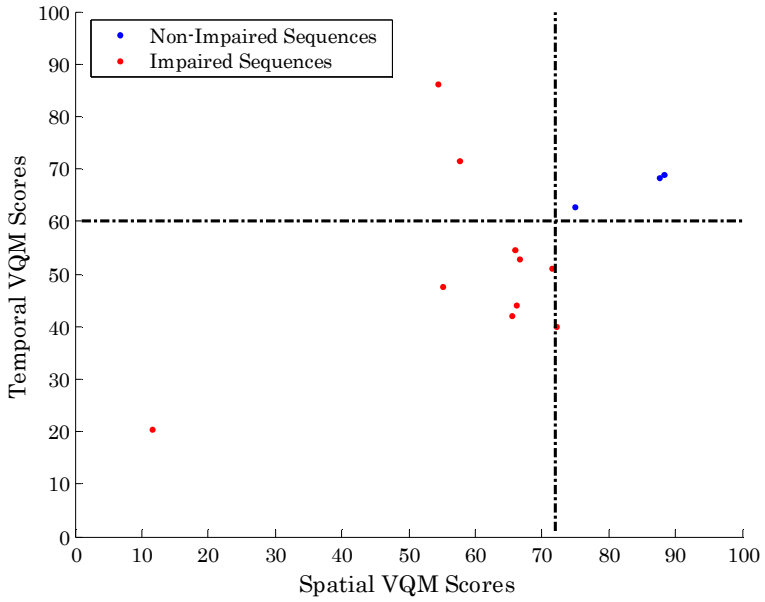


Fig. 93 – Threshold selection for rigid texture scenario

The quality scores obtained from the proposed global VQMs for non-rigid textures are given in Fig. 94. It can be seen that, in this scenario, a clear distinction between good and bad synthesis results can be achieved for temporal artifacts. This shows that the proposed global temporal VQM is efficient for corresponding artifact detection given non-rigid synthetic textures. Spatial artifacts are

however more difficult to separate. A discrimination threshold can be chosen in the interval $[68\ 74]$ for spatial artifacts, while it should lie in the interval $[68\ 77]$ for temporal artifacts. This can be verified in Fig. 94 and Fig. 95.

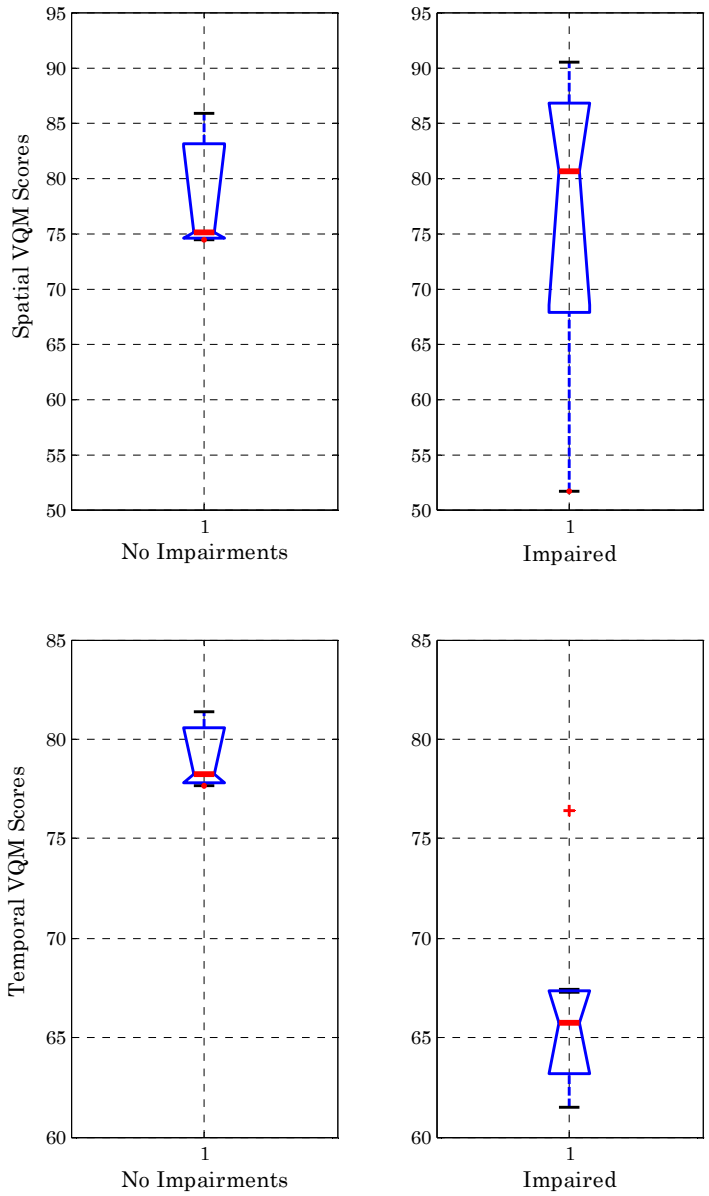


Fig. 94 – Quality scores for synthetic video sequences with and without noticeable impairments (non-rigid textures)

A synthetic result is considered to be impaired if one of the thresholds is undershot. The classification results obtained for the test sequences are given

in Tab. 13. The spatial threshold is set to 70, while the temporal threshold is set to 75.

Sequence	Subj. Score	Q_S	Q_t	Output
Canoe – A	Bad	83,64	61,5	IMPAIRED
Canoe – B	Bad	86,75	67,36	IMPAIRED
Ducks	Good	74,44	81,39	OK
Flood	Bad	90,47	64,31	IMPAIRED
Rain	Bad	51,66	76,4	IMPAIRED
Shuttle	Bad	77,59	63,18	IMPAIRED
Synchronized Swimming	Good	85,79	78,24	OK
Whale	Good	75,05	77,66	OK
Whale – NoAlignment	Bad	67,89	67,19	IMPAIRED

Tab. 13 – Classification results for non-rigid textures

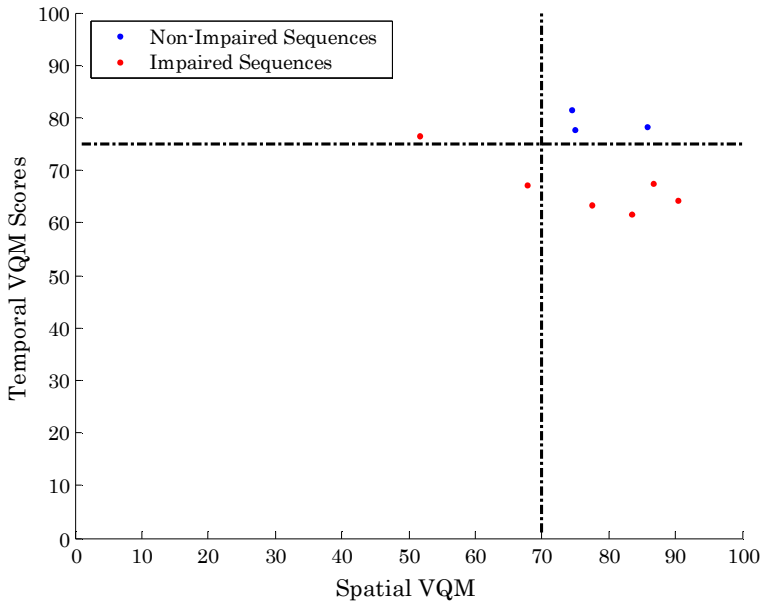


Fig. 95 – Threshold selection for non-rigid texture scenario

It can be seen in Fig. 93 and Fig. 95 that the distribution of the synthetic sequences in the objective VQM space differs for rigid and non-rigid textures. Most of the examined sequences feature both spatial and temporal artifacts for rigid texture synthesis, while non-rigid synthesis mostly yields only temporal artifacts. These clues might indicate directions for improvement of the texture synthesis algorithms.

6.7 Discussion

The global quality assessment tools presented in this paper represent a trade-off between low complexity and thus low perceptual correlation on the one hand and high complexity and thus high perceptual correlation on the other hand. In fact, the proposed global spatial VQM is more than a compromise as it always yields a slightly higher correlation coefficient (Spearman) than the best quality measure by VQEG. The proposed measures can be easily extended to other applications than coding. This is achieved through adequate feature point selection.

Only the global VQMs have been evaluated in this chapter. The local quality measures are addressed in the experimental evaluation of the overall system in Sec. 10.3.4. This is justified by the fact that comparison of the proposed VQM to the state-of-the-art requires global quality evaluation, as no mask correction is required in the test framework by VQEG [213]. Furthermore, in the video sequence classification experiments, mask correction is not required to achieve meaningful statements.

7 State Machine

A detailed insight into the interaction between the modules of the proposed content-based video coding scheme is provided in this chapter. The task of the state machine consists in exploring relevant degrees of freedom of the analysis-synthesis loop (cp. Fig. 2), where the relevance notion refers to system states that yield significant variance of the generated content-based side information or, alternatively, to states that are known to alleviate typical, sporadic distortions caused by given system components. The most influential modules, under this premise, are the texture analyzer and the texture synthesizer for rigid textures, whose states of interest are described in the following sections.

7.1 Texture Analyzer States

The texture analyzer features two major options that are systematically explored by the state machine. They are the spatial and the spatio-temporal analysis (cp. Secs. 4.1 and 4.2). This can be explained by the fact that, if fundamental assumptions of the perspective motion model or the optical flow estimator are violated by the motion properties of the given video clip, motion analysis becomes unreliable (e.g. non-rigid texture as water in the video). In such cases, spatial coherence criteria as color or texture are applied to cluster picture samples. It is hence indicated to generate the spatial and spatio-temporal analysis options for an a posteriori evaluation of their performance.

7.2 Texture Synthesizer States

The texture synthesizer for rigid textures operates intra-synthesis for samples that are not available in the reference pictures (cp. Sec. 5.2). Causes of unavailability are thereby twofold. They relate to covering and uncovering effects and to properties of the segmentation masks generated by the texture analyzer. That is, samples of a detail-irrelevant texture segment are considered invalid if they lie outside the corresponding region in both reference pictures. Hence, invalid samples typically occur at picture (cp. Fig. 96, “Husky” picture) or texture borders (cp. Fig. 96, “Flower Garden” and “Husky” pictures).

Intra-synthesis yields imperceptible distortions if the invalid samples represent thin line structures or “small” blobs as depicted in Fig. 96 (cp. green ellipses in pictures).

Fig. 96 – Success and failure of the intra-synthesis method. Picture from “Flower Garden” test sequence with invalid samples at texture border marked white (top left), “Flower Garden” picture with intra-synthesized invalid samples (top right), picture from “Husky” test sequence with invalid samples at picture and texture borders marked white (bottom left), “Husky” picture with intra-synthesized invalid samples (bottom right).

Visible distortions are however obtained for large blobs of invalid samples as can be seen in Fig. 96 (cp. red ellipses in pictures). The annoyance of these distortions is strongly correlated with the properties of the given texture. Textures for which the stationarity is not captured by the dimension of the considered neighborhood may not be successfully synthesized by the intra-synthesis approach (cp. Sec. 5.2.3.2).

The rigid texture synthesizer options can now be summarized as follows:

1. Use full mask, including all invalid samples, for synthesis,
2. Use mask, without "large" invalid sample blobs, for synthesis,
3. Use mask, without any invalid sample, for synthesis.

The first option potentially yields the largest distortions, while the second one entails smaller impairments. The option with the lowest distortions is finally option 3. Notice that "large" blobs are determined using a threshold defining the tolerable amount of invalid samples within a macroblock, i.e. 100% for option 1, 12.5% for option 2 (max. 2 pixel thick stripes) and 0% for option 3 in this thesis.

7.3 State Diagram

The state diagram of the analysis-synthesis loop is depicted in Fig. 97. It can be seen that the spatial texture analysis (STA) is selected first (transition 0). The identified texture regions are then post-processed by the side information generator for rigid textures (SI_r, transition 1). The generated (quantized) side information is used for texture synthesis (TSI, transition 2, cp. also Sec. 5.2). The result of the latter operation is submitted to the video quality assessor (transition 3, cp. also Sec. 6.4) that modifies the segmentation masks provided by the texture analyzer at impaired spatio-temporal locations. That is, macroblocks that feature noticeable distortions are rejected from synthesis and coded by the fallback encoder. The modified side information resulting from video quality assessment is transmitted to the rate-distortion decision module (transition 4). Two further TSI options are executed using the initial spatial segmentation masks. They yield the transitions 6, 7, 8 (option 2 in Sec. 7.2) and 10, 11, 12 (option 3) respectively. As depicted by the dotted transition from VQA to SI_r, each TSI option is iteratively executed until good synthesis quality is attained or the maximum number of iterations has been reached. In the latter case, the modified detail-irrelevant segment is totally rejected as synthesis failed. Note that the transitions corresponding to iterative calls of TSI options are not depicted in Fig. 97 for legibility reasons.

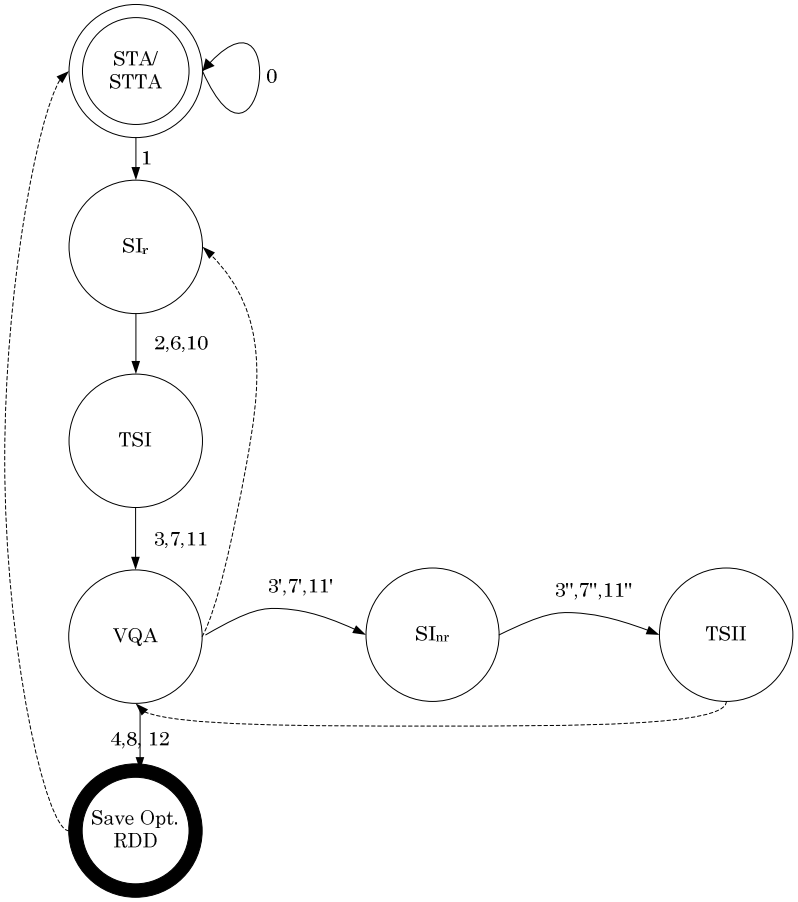


Fig. 97 – State diagram of the closed-loop analysis-synthesis video coding algorithm for a group of pictures

The VQ assessor invokes the texture synthesizer for non-rigid textures (TSII) in case TSI fails to synthesize a given texture region in a GoP (transitions 3', 3", 7', 7", 11', and 11"), where the "prime transitions" are derived from the corresponding TSI transitions. TSII collects the genuine segmentation masks of the spatial texture analysis module (TSI states are ignored) until the required bursts are filled (cp. Sec. 5.3). The given texture is then synthesized. It is rejected in case it is not visible over an adequate time interval in the video sequence. It is obvious that TSII must be able to handle a set of regions simultaneously, which has a significant impact on memory requirements. Notice that TSII is operated using spatial masks because the spatio-temporal analyzer is not tuned for non-rigid texture identification. The assumption here is that non-rigid textures may be more efficiently captured by applying a homogeneity criterion distinct from

motion. Iterative call of TSII can be conducted similarly to TSI to improve synthesis quality as depicted by the dotted transition from TSII to VQA.

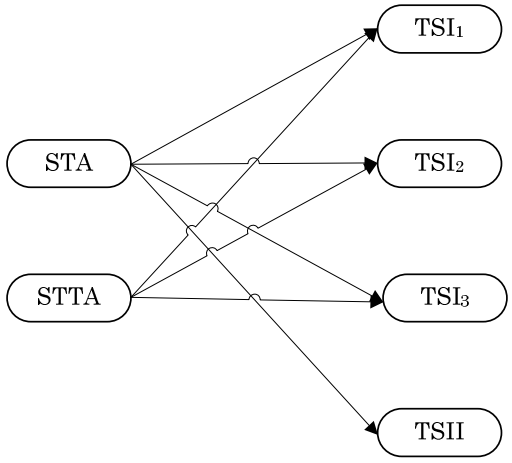


Fig. 98 – Synthesis options selectable by texture analysis options

The dotted transition from the rate-distortion decision module to the texture analyzer switches between spatial (STA) and spatio-temporal (STTA) analysis, once all TSI states have been evaluated. The states that can be reached by the STA and the STTA are depicted in Fig. 98. As can be seen, the TSII option can not be selected when STTA side information is active. This relates to the fact that the STTA identifies only rigid textures. The TSII side information is merged with the six TSI side information sets (3 STA-related options and 3 STTA-related options), where the TSI regions overrule TSII regions. Hence, the TSII regions are modified before the VQA-SI_{nr}-TSII loop is activated. Note that overlap between TSI and TSII regions can only occur in the STTA-related options, where the rigid textures areas are assumed to be particularly reliable.

8 Rate-Distortion Optimization

In this thesis, the rate-distortion optimization problem consists in identifying the content-based side information that yields minimized distortion of the decoded video sequence under consideration of bit budget constraints. The side information options thereby correspond to the degrees of freedom of the spatio-temporal texture analysis module (cp. Chapter 4) and the texture synthesizers (cp. Chapter 5).

8.1 Previous Work

In the framework of lossy compression, high compression ratios are achieved at the cost of source signal fidelity. This information-theoretic problem, also called Rate-Distortion (RD) problem, was initially formulated by Shannon in [216],[217]. The corresponding theory addresses the problem of minimization of the number of bits allocated to a source signal given an upper distortion bound of the latter. The rate-distortion trade-off has been a very active research field during the last decades [218]. Some theoretic performance bounds have been derived in the literature. They give the reachable and non-reachable points in the rate-distortion trade-off [218],[4], albeit for simple statistical source signals as independently identically distributed scalar sources with Gaussian, Laplacian, or generalized Gaussian distributions. In fact, knowledge of statistical properties of the source is required to determine performance bounds, which is a formidable task given complex sources as video sequences.

Although theoretical performance bounds provide useful information, they are typically not tight enough for practical applications. This relates to the fact that any lossy data compression scenario features a finite set of admissible quantizers, which implies that only a limited number of RD pairs is available for any given source. The latter pairs define the so-called Operational Rate-Distortion Function (ORDF), whose convex hull gives the boundary between practically and theoretically reachable performance [218],[219],[4]. Notice that the rate-distortion optimization performance is inherently limited by the operating points achieved through the coding framework, as the best RD pair is selected among the set of data pairs.

Ortega and Ramchandran [218] enumerate three aspects that are to be considered before conducting an RD optimization. These are the selection of the basic coding unit, the complexity, and the cost function. The basic coding unit corresponds the granularity of the RD optimization process. The former can be a pixel, a macroblock, an object, or a full, rectangular image. The complexity issue relates both to processing and memory complexity. Processing complexity is given by the fact that several coding/decoding operations must typically be conducted in order to obtain the required RD pairs, except an adequate ORDF model is available. A quest for the best RD pair must also be conducted at this processing step. Memory complexity is closely linked to delay matters and is due to the required buffering of RD pairs for the search of the optimal solution. The cost function is determined for each coding unit and involves simultaneous optimization of the bit rate and the distortion.

It has been shown by Everett [220] that the RD optimization problem can be seen as the minimization of

$$\mathrm{J} = D + \lambda R \tag{133}$$

where J is the Lagrangian cost function [4],[218],[221]. D represents the distortion, R the corresponding bit rate, and λ the Lagrangian multiplier. The latter is used to determine the extremal points of a function of one or more variables subject to one or more constraints. The Lagrangian multiplier method can only be used to identify the points on the convex hull of the ORDF [218],[219],[4].

Several rate-distortion optimization techniques for region-based coding approaches have been proposed in the literature, e.g. [9],[222]. In [9], a partition tree is built. A bottom-up analysis is applied to identify the best segmentation mask. It is evaluated if a region R is to consider as a whole or if rate-distortion properties can be improved by partitioning it into several sub-regions R_n. This is done under usage of (133). The distortion is thereby assumed to be additive over the regions R_n. That is, the Lagrangian of R is compared to the sum of the Lagrangians of R_n. If the latter is smaller than the former, a split representation

of R is selected. R is maintained otherwise. The approach presented in [9] is iterative and halts at the root node of the partition tree. The optimum λ is obtained through a gradient search algorithm. Wang et al. [222] propose an RD optimal bit allocation scheme for object-based video coding. Their approach is based on Lagrangian relaxation and dynamic programming. Each object is thereby processed separately, where corresponding shape, texture and motion information is jointly optimized. The bisection method [219] is used to determine the optimal Lagrangian multiplier.

8.2 Proposed Rate-Distortion Optimization Approach

The basic coding unit in the present content-based video coding framework is a spatio-temporally consistent region. Consistency criteria are formulated by the texture analyzer in Sec. 4.2.4. Given such a region, RD pairs are calculated for all relevant settings of the texture analyzer and the texture synthesizers. The distortion of the source resulting from the operated analysis-synthesis is measured as defined in Sec. 6.4. The amount of bits required to encode the side information generated by the present approach is determined by the encoder.

8.2.1 Overall Approach

The rate-distortion optimization procedure proposed in this thesis corresponds to a two step approach. Firstly, for each valid side information option, the RD costs are evaluated against the reference codec for each detail-irrelevant texture. In the second step, given the RD-optimized content-based side information options, rate-distortion decisions are made GoP-wise. That is, in the selected optimal side information option O_{opt}, all detail-irrelevant regions are generated using the same analysis-synthesis loop configuration. The RD optimization algorithm is presented in detail in the following section.

8.2.2 Rate-Distortion Decision Criteria

In the present rate-distortion framework, it is assumed that the video quality measure proposed in Sec. 6.4 entails detection and rejection of subjectively annoying impairments. This hypothesis is justified by the fact that high rank correlation coefficients are achieved for the VQM (cp. Sec. 6.6). That is, the quality ranking predicted by the VQMs show a very high correlation to the perceptual quality ranking of a given data set. Hence, given a set of quality

estimates related to side information options, O_1 … O_n, the former are sorted and the thresholds determined in Sec. 6.6 are applied. The validated options, O_k … O_m, are assumed to feature subjectively imperceptible distortions. This is depicted in Fig. 99, where n side information options are evaluated by the VQMs and a subset $m-k+1$ (with $1 \le k \le m \le n, k, m, n \in$ IN) of these are validated.

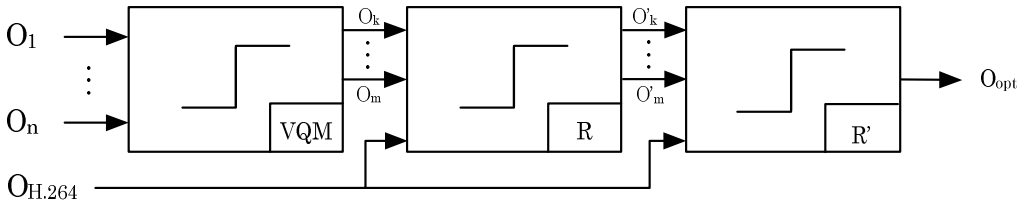


Fig. 99 – Rate-distortion optimization approach for the content-based video coding framework

The validated options may, however, contain low frequency textures that can be effectively represented by the reference video codec, such that it is not recommended to schedule that type of texture for synthesis. Hence, for each validated option, O_k … O_m, and each potentially detail-irrelevant texture, a comparison against the reference codec is conducted on a best rate selection basis. This yields edited side information options, O'_k … O'_m, with improved coding gains in terms of bit rate. For instance, assuming that a mask generated by the texture analyzer features a low frequency texture, the latter will be rejected by the best rate selection operation described above. That is, the corresponding macroblocks will be coded by the reference codec. This is shown in Fig. 99, where the validated options as well as the fallback codec option are examined by the second module depicted. Note that the latter module only applies to TSI regions, as no picture-by-picture evaluation can be made for non-rigid textures.

Finally, the optimized content-based side information options, O'_k … O'_m, are submitted to a second best rate selection operation. The option (reference codec included) yielding the lowest overall bit requirements for a given GoP is selected as the optimal option O_{opt}. This can be seen in Fig. 99, where the second best rate decision is referred to as R'. This approach guarantees equal or smaller bit rates than the reference codec.

8.3 Discussion

The proposed RD optimization approach is viable but does not provide a joint optimization of rate and distortion variables. Increased performance is expected from such a strategy. The latter is, however, beyond the scope of this work and is let to future improvements of the proposed coding scheme.

Dependency problems are not solved in this thesis [218]. They relate to the fact that spatial as well as temporal dependencies exist between macroblocks in H.264/MPEG4-AVC. This is due to motion compensated predictive and differential coding. Texture synthesis yields such dependencies due to motion compensation (translational model for non-rigid texture synthesis and perspective model for rigid texture synthesis). These dependencies hamper the RD optimization task. Hence, in this thesis, the independency approximation is preferred to speed up computation [218]. Nonetheless may the consideration of given dependencies increase RD optimization performance.

O_{opt} may be generated by cross-option decisions with major compression gains. Let's assume that O_1 holds regions R_{11} and R_{12}, and that O_2 holds regions R_{21}, R_{22}, and R_{23}. It is further assumed that R_{11} and R_{21} are of different sizes but hold the same texture. Further assuming that O_2 yields the best RD performance for the considered texture, R_{21} and corresponding parameters are copied to the O_{opt} buffer. R_{12} is now processed, but the correspondence to R_{22} and R_{23} is assumed unclear because these regions overlap both R_{11} and R_{12} and contradicting texture labels are given due to the diverging homogeneity criteria (color for O_1, motion for O_2) of the texture analysis options. It is obvious that in such (frequent) cases a one-to-one cross-option region assignment might become very complex. It is, however, expected that major additional gains are achievable here if this drawback is tackled.

9 System Integration into H.264/MPEG4-AVC

In this chapter, the integration of the proposed video coding algorithm into an H.264/MPEG4-AVC codec is presented. Required decoder syntax adaptations as well as encoder-sided signal processing requirements are described. To begin with, a short overview of existing digital video coding standards is given. The advantages of H.264/MPEG4-AVC compared to other standards are highlighted and its most important properties are described.

9.1 Overview

Several digital video coding standards have emerged during the past 15 years. They have the benefit of ensuring interoperability among products of different manufacturers. The most important standards have been developed by the Video Coding Experts Group (VCEG) of the ITU-T and the Motion Pictures Experts Group (MPEG) of ISO/IEC. Well-known VCEQ standards are H.261 [223], H.262 [224], and H.263 [225], while MPEG standards are known as MPEG-1 [226], MPEG-2 [224], and MPEG-4 [1]. These video standards are all block-based hybrid coding methods. That is, they exploit spatial as well as temporal statistical dependencies of the source signal. It should be noted that H.262 and MPEG-2 relate to the same standard. The latter was developed as a result of the first collaboration between VCEG and MPEG. The MPEG-2 standard is of major importance for digital television systems world wide (satellite, cable, and terrestrial emission) and the most widespread video standard today.

Due to the emergence of new services (e.g. video transmission over xDSL or UMTS) and the rising acceptance of high definition television, there is a need for higher compression efficiency than can be provided by the video standards cited above. E.g. given an increased coding efficiency, more TV channels can be transmitted by broadcasters. Hence, VCEG and MPEG joined their efforts within the Joint Video Team (JVT) to release a new video standard called H.264/MPEG4-AVC [3]. This video coding standard is the most efficient to date and was designed to reduce MPEG-2's bit rate by a factor of two on average. H.264/MPEG4-AVC accounts for requirements related to network-based applications through adequate video representation. Conversational (video

telephony) as well as non-conversational (storage, broadcast, or streaming) scenarios are considered [227]. The important modifications of the new standard relate to details of each functional element of a hybrid video codec. Similarly to preceding video codecs, H.264/MPEG4-AVC standardizes only the bitstream syntax and the decoding process. This ensures the required flexibility of the standard by allowing enough room for subsequent application-optimized implementations.

The H.264/MPEG4-AVC design features two major layers, namely the Video Coding Layer (VCL) and the Network Abstraction Layer (NAL). The former layer ensures efficient video representation, while the latter layer formats the VCL data for transmission or storage.

9.1.1 Video Coding Layer

The VCL is depicted in Fig. 100. An input picture is segmented into rectangular macroblocks. The latter are spatially clustered into slices that can be decoded independently. Slices are however not necessarily composed of connected macroblocks. Macroblocks are coded using either the inter or the intra mode. For macroblocks coded in the intra mode, a given prediction macroblock is determined using only information from the currently processed macroblock, i.e. from spatially neighboring samples. In the inter mode, a given prediction macroblock is determined through motion-compensated prediction using one or more previously decoded pictures as reference(s). Motion information is transmitted to the decoder in this mode. So-called Predictive (P) and Bi-predictive (B) slices support both intra and inter coding modes. Given display order, B and P slice macroblocks can be predicted from temporally preceding as well as succeeding reference pictures. The major difference between the two slice types resides in the way the prediction signal is built from the reference pictures. A P or B slice macroblock can be coded in the SKIP mode that is characterized by the fact that neither the quantized prediction error, nor a motion vector or a reference index parameter are transmitted. Notice that the reference index parameter signals which of the reference pictures located in the reference picture buffer should be used for prediction. The SKIP mode is a very efficient coding method that is selected if the properties of a macroblock are such that, firstly, its motion vector can be predicted from neighboring (spatial) or co-located (temporal)

macroblocks and, secondly, it contains no non-zero quantized transform coefficients [228]. Further slice types are defined in the standard [3] and described in [228],[227]. The first picture of the sequence as well as random access points are typically completely intra coded, while the remaining pictures are usually at least partially inter coded.

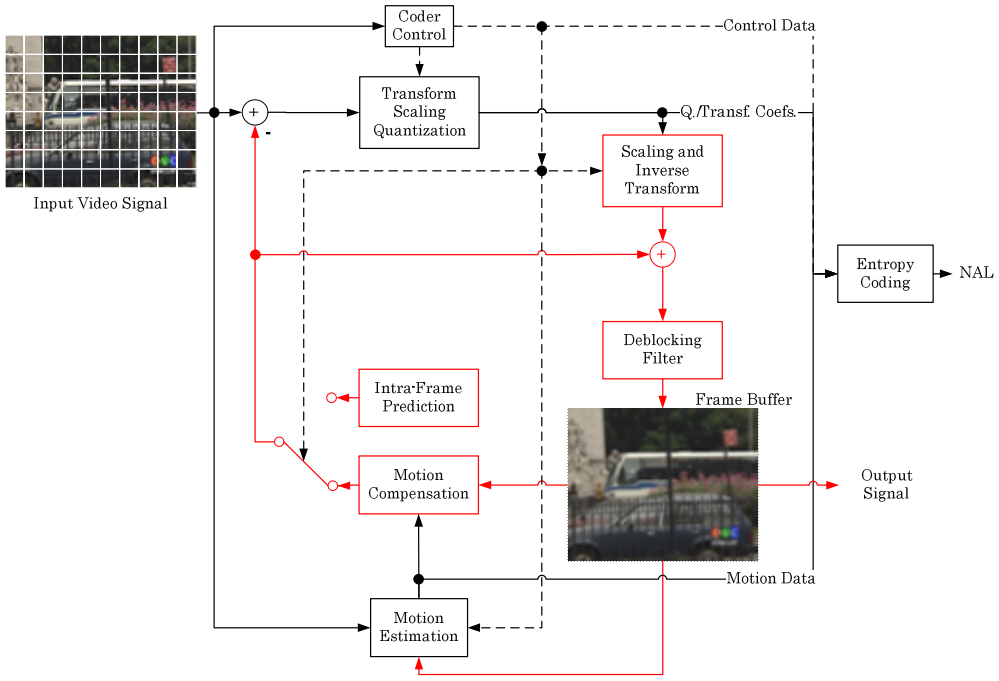


Fig. 100 – Block diagram of the H.264/MPEG4-AVC encoder for a macroblock [227] with highlighted (red) decoder path

Usually, at the encoder side, the predicted macroblocks are subtracted from the original macroblocks to give the residual error signal. The latter is transformed and quantized to give a set of quantized transform coefficients. A separable integer transform is used in the H.264/MPEG4-AVC standard instead of a Discrete Cosine Transform (DCT). The former transform features similar properties as the 4x4 DCT. Inverse transform inaccuracies are avoided by definition of an exact integer inverse operation. Scalar quantization is used to quantize the transform coefficients. They can be quantized using 52 quantizers that are selected via a Quantization Parameter (QP) on a macroblock basis.

The transform coefficients are entropy coded for transmission to the decoder. Entropy coding methods are lossless. They assign data elements a respective coded representation, whose efficiency in terms of data size can be improved by consideration of context information as anterior predictions, transformations, and quantization. Two entropy coding methods are used in the H.264/MPEG4-AVC video coding standard, namely Variable Length Coding (VLC) and Binary Arithmetic Coding (BAC). Both approaches are used in a Context Adaptive (CA) manner yielding the so-called CAVLC and CABAC entropy coding methods [228]. CABAC typically provides significant bit rate reduction compared to CAVLC. This relates to the fact that CABAC as any arithmetic coding technique allows the assignment of non-integer number of bits to each symbol of an alphabet. CABAC also features a more efficient context-adaptive scheme than CAVLC [227]. The efficiency of CABAC is however achieved at the cost of an increased complexity compared to CAVLC.

The encoder features a recursive reconstruction path, highlighted in red in Fig. 100, that corresponds to an inbuilt decoder. Hence, the encoder mimics the decoder operations, i.e. the quantized transform coefficients are rescaled and inverse transformed yielding the decoded prediction residual [227]. The latter is added to the prediction signal to give the reconstructed signal. A so-called deblocking filter is finally applied to alleviate blocking artifacts in the reconstructed video. The reconstruction path in block-based hybrid video codecs ensures that the reference pictures or macroblocks used for temporal or spatial prediction are identical at the encoder and at the decoder side. This in turn guarantees a synchronous processing of the video signal on both sides.

9.1.2 Network Abstraction Layer

The NAL formats the VCL data and generates a generic header that conforms to both packet-oriented and bitstream system requirements. The VCL data are confined into NAL units that contain an integer number of bytes each. Several NAL unit types, as the Picture Parameter Set (PPS) and the Sequence Parameter Set (SPS), are defined in the H.264/MPEG4-AVC standard. The SPS represents a syntax structure containing syntax elements that apply to entire coded video sequences, while the PPS contains syntax elements that apply to the entire coded pictures [3]. Other important NAL units are the Slice Header (SH) and the SEI

messages, where the SH is actually a part of a NAL unit. The SH is a part of a coded slice containing the data elements relating to the first or all macroblocks represented in the slice. SEI messages assist in processes related to decoding, display or other purposes. However, SEI messages are not required for constructing the luma or chroma samples by the decoding process [3].

9.2 System Integration

System integration consists in the integration of the proposed analysis-synthesis loop into the encoder and the incorporation of the texture synthesizers into the decoder. Efficient coding of the content-based side information, as well as signaling of the macroblocks to be synthesized are further required.

The integration of the texture analyzer into the encoder requires delay buffers to conduct a look-ahead, content-based video evaluation in order to operate a synthesis mode decision. The first delay buffer, T_1 (cp. Fig. 101), collects an entire GoP before the analysis-synthesis loop, referred to as ASL in Fig. 101, is called. Thorough content analysis is then conducted. A second delay buffer, T_2 (cp. Fig. 101), retains the incoming GoPs until the required amount of pictures is available for coding. The length of T_2 depends on the texture properties and is basically dictated by the texture synthesizer for non-rigid textures. It is remembered here that rigid texture synthesis is conducted GoP-wise, while several GoPs are required to operate non-rigid texture synthesis (cp. Secs. 5.2.1 and 5.3.4). Good synthetic results can only be achieved for non-rigid textures if the reference bursts are "large enough" to capture the stationarity of the texture and the scale of the texture elements is known as explained in [144] and Sec 5.1. The GoP length depends on the specific motion properties of the given video sequence in the rigid texture synthesis case (cp. Sec. 5.2.1). Once the required amount of pictures is reached in T_2, one GOP is coded. T_2 is then refilled etc..

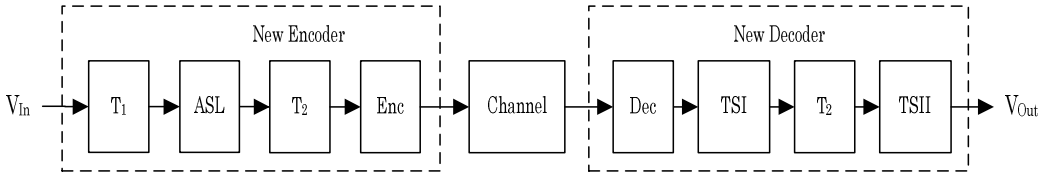


Fig. 101 – Integration of the analysis-synthesis method into the H.264/MPEG4-AVC video codec

For coding, different content-based side information sets and the default coding mode (H.264/MPEG4-AVC) are considered. The rate-distortion properties of these options are now to be evaluated. For that, each GoP is successively coded with the available side information options. For each GoP, the best option is selected. Fig. 102 depicts the iterative coding process for a macroblock and a single side information option. It can be seen that based on a segmentation mask, a given macroblock is coded either using H.264/MPEG4-AVC or the analysis-synthesis method. A flag, referred to as CodeFlag in the following, is required to signal the encoder, if all side information options have already been coded. Is this not the case, then the generated macroblock symbols are saved together with the bit costs in an RDD buffer that serves as input to the RDD module (cp. Fig. 99). The latter symbols are written to the bitstream otherwise. This is repeated for all macroblocks of a given slice, for all slices of a given GoP, and for all options of a given GoP. Note that the CodeFlag is set after rate-distortion decision has been conducted. Such that the best option is already known at that stage and written to the bitstream.

Browsing through the available set of side information requires careful handling of the reference picture buffer. The latter is a first in first out (fifo) buffer of finite size. Is the buffer already full and a further reference picture added to it, then the oldest reference picture is dropped from the buffer. Let's assume this happening for a given GoP. The initial state of the reference picture buffer must then be restored before examining the next side information option. Inappropriate handling of the fifo buffer will yield encoder-decoder inconsistencies due to diverging states of the fifo buffer. This requires an increased amount of memory as for each picture that falls out of the fifo, the picture itself, the motion vectors and macroblock symbols (mode, reference picture, etc.) are retained.

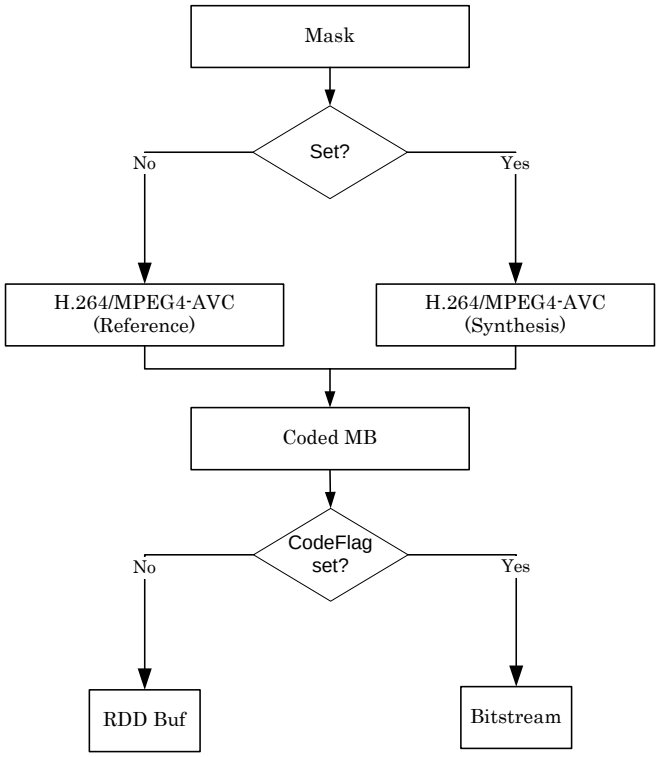


Fig. 102 – Browsing through content-based side information options for rate-distortion optimization of a macroblock

The side information of the analysis-synthesis framework (cp. Tab. 3 and Tab. 4) is transmitted to the decoder via a new SEI message. Hence, the bit-stream is fully decoded by the standard conforming decoder. While decoding, all macroblocks belonging to a synthesizable texture region are handled as skipped macroblocks. Texture synthesis is then operated and all reconstructed YUV samples of macroblocks belonging to a synthesizable texture region are replaced. It can be seen in Fig. 101 that the synthesizer for rigid textures (TSI) is called after the standard conforming decoder. Output pictures of the rigid texture synthesizer are stored in a delay buffer if at least one non-rigid texture is to be synthesized. Once all pictures of the corresponding reference and synthesis bursts are released, TSII synthesis is carried out (cp. Sec. 7.3).

9.2.1 Analysis-Synthesis Information SEI Message Syntax

The syntax of the SEI message for the analysis-synthesis information is given in Tab. 14. This information relates to a single picture or slice in the given framework. The syntactical description presented in Tab. 14 conforms to the specifications in the H.264/MPEG4-AVC standard [3].

analysis_synthesis_info(payloadSize) {	C	Descriptor
copy_marker	5	u(1)
burst_marker	5	u(4)
num_detail_irrelevant_textures	5	ue(v)
num_motion_parameter_sets	5	ue(v)
num_splits	5	ue(v)
for(i = 0; i < num_detail_irrelevant_textures; i++) {		
control_flags[i]	5	u(2)
}		
for(i = 0; i < num_motion_parameter_sets; i++) {		
motion_parameters[i]	5	se(v)
}		
for(i = 0; i < num_splits; i++) {		
split_order[i]	5	ue(v)
}		
while(current_state==mask_with_detail_irrel_textures) {		
mb_in_binary_mask_with_same_label++	5	ue(v)
}		
}		

Tab. 14 – Syntax of the SEI message for the analysis-synthesis information

Category C is a number associated with each syntax element. The former specifies the partitioning of slice data, where category five relates to SEI messages. Slice data partitioning corresponds to a method of partitioning selected syntax elements into syntax structures [3].

Descriptors specify the parsing process of each syntax element (cp. Tab. 14). They are:

- se(v): Signed integer Exp-Golomb-coded [4] syntax element with the left bit first. The parsing process for this descriptor is specified in [3].
- u(n): Unsigned integer using n bits. When n is "v" in the syntax table, the number of bits varies depending on the values of other syntax elements. The parsing process for this descriptor is specified by the return value of

the function read_bits(n) interpreted as a binary representation of an unsigned integer with most significant bit written first [3].

- ue(v): Unsigned integer Exp-Golomb-coded syntax element with the left bit first. The parsing process for this descriptor is specified in [3].

Only relevant descriptors, for the given side information, are described here. An exhaustive list of the descriptors can be found in [3].

9.2.2 Analysis-Synthesis Information SEI Message Semantics

The semantics of the side information of the proposed analysis-synthesis video coding approach are described in Tab. 15.

Syntax Element	Semantics
copy_marker	Flag indicating that reference pictures of given picture are to be buffered. Required for hierarchical B picture coding [229] for unambiguous reference picture identification Cp. Tab. 4
burst_marker	Indicator for picture's affiliation to ref or synth burst (cp. Tab. 3 and Tab. 4)
num_detail_irrelevant_textures	Number of rigid and/or non-rigid detail-irrelevant textures in given mask
num_motion_parameter_sets	Number of motion parameter sets available for description of rigid and/or non-rigid detail-irrelevant textures
num_splits	Absolute number of regions in mask. The number is greater than zero only in case at least one texture region is split
control_flags	Key picture with highest priority (cp. Tab. 3)
motion_parameters	Cp. Tab. 3 and Tab. 4
split_order	Labels of texture segments in segmentation mask. Transmitted only if "Split Flag" set (cp. Tab. 3 and Tab. 4)
mb_in_binary_mask_with_same_label	Maximum number of succeeding macroblocks, in binary mask showing detail-irrelevant textures (mask_with_detail_irrel_textures), with the same label (cp. Tab. 3 and Tab. 4). Macroblock counting is stopped, when a label change is observed in the mask

Tab. 15 – Semantics of the SEI message for the analysis-synthesis information

Note that the quantization parameters for the motion parameters are transmitted via the sequence parameter set. It is, however, possible to transmit the quantization parameters via the proposed SEI message to enable picture-adaptive quantization decisions.

10 Experimental Results of Overall Framework

The proposed content-based video coding approach has been integrated into an H.264/MPEG4-AVC codec. Test sequences are used to demonstrate that an approximate representation of some rigid and non-rigid textures can be done without subjectively noticeable loss of quality. The contribution of the modules of the proposed framework to the overall efficiency is also documented in the following.

10.1 Ground truth

The ground truth set consists of three test sequences. Two of these are well-known and often used to assess video coding algorithms, i.e. “Flower Garden” and “Concrete”. Both sequences contain rigid textures as can be seen in Fig. 72 and is given in Tab. 9. The remainder sequence is “Sea”. It shows non-rigid, namely water texture, as can be seen in Tab. 5. All test sequences are progressive color clips of up to 10s, i.e. 300 pictures at 30Hz frame rate.

10.2 System Configuration

The selected configuration of the proposed video codec is described in this section. It is thereby distinguished between the H.264/MPEG4-AVC specific settings and the configuration of the analysis-synthesis loop.

For the H.264/MPEG4-AVC codec, one reference picture for each P picture, CABAC (entropy coding method), and rate distortion optimization are set. Schwarz et al. have shown in [229] that hierarchical B picture coding significantly enhances H.264/MPEG4-AVC coding efficiency. For that, three hierarchically structured B pictures are used in the experiments: $IB_2B_1B_2PB_2B_1B_2P...$ The quantization parameter is set to QP=16, 20, 24, 28, 32, 36, and 40. P and I pictures are coded with QP+1, while the B_1 pictures are coded with QP+5 and the B_2 pictures are coded with QP+6.

The spatio-temporal texture analyzer used in the analysis-synthesis loop is configured as given in Tab. 2. Spatial segmentation masks are not considered for rigid texture synthesis. Furthermore, TS_r is used in the safe mode, where only

samples are synthesized that are available in the reference pictures (cp. Sec. 7.2). A search range of 18x5 (abscissa x ordinate) is used for TS_r at CIF (352x288) resolution, while a range of 36x10 is used for SD (720x576) and 720p (1280x720) resolutions (cp. 5.2.2). The configuration of TS_{nr} is given in Sec. 5.3.5.2. Reference bursts are assigned a length of 20 pictures, while synthetic bursts have a length of 70 pictures. The size of the delay buffer, T_2, is thus set to 110 pictures in our experiments (cp. Sec. 9.2). For the video quality assessor, following thresholds are used: A cluster of 12 pixels is required for a synthetic texture neighborhood to be considered as suspect (cp. Sec. 6.4.4). The threshold for histogram matching in the local quality assessment context is set to 0.15, while the global thresholds (cp. Fig. 61) are set as given in Sec. 6.6.4.3.

10.3 Performance and Properties of Overall Framework

10.3.1 Rate-Quality Performance

The video coding approach presented in this work is based on the idea that some textures can be approximately reconstructed at the decoder without entailing subjectively annoying artifacts. Hence, the MSE-based PSNR measure is not an adequate distortion criterion for the approach under test. For that, rate-distortion curves are not used for performance assessment of the proposed system. Following these observations, subjective evaluations have been conducted to achieve a rate-quality curve, which is more likely to provide in-depth insight into the potentialities of the proposed approach.

The triple stimulus continuous evaluation scale method (cp. Sec. 6.3.5.4) is used for subjective evaluations. It is, in fact, expected to allow comparison of H.264/MPEG4-AVC and the proposed codec in an unbiased way. For practical reasons, instead of three displays, a single LCD/TFT display (DELL UltraSharp 2407FPW) , with 1920x1200 pixel resolution and a 24 inch diagonal, is used to present the sequences under test. The display is calibrated with the Spyder2PRO calibration system by COLORVISION®. The display is used in the portrait mode, where three SD sequences are displayed simultaneously. The lower picture quality anchor is selected as the analysis-synthesis outcome at the lowest bit rate or respectively the highest coding quantization parameter. This is justified by the fact that the latter sequence type potentially contains both relevant artifact

classes, i.e. block coding artifacts due to H.264/MPEG4-AVC and artifacts related to synthesis.

31 test subjects are asked to evaluate the performance of the proposed system. 9 of the test subjects are women, while the remainder 22 are men. The test subjects are aged 18 to 47. The distance of the test subjects to the display is selected as 4 times (SD resolution) and 8 times (CIF resolution) the test video's height, which corresponds to comfortable viewing conditions in the given setup. Before evaluations are conducted, each test subject is explained the experiment in an individual training session, where she or he is also given the opportunity to ask questions and to evaluate a video testwise. Each video sequence is evaluated on a scale from 0 (bad) to 100 (excellent).

All test sequences are evaluated in this test, where "Flower Garden" and "Concrete" are shown at SD resolution, while "Sea" is shown at CIF resolution. For each sequence, four QP levels (16, 24, 32, and 40) are selected. Such that each test subject is called upon to evaluate 8 video clips per test sequence, i.e. four clips per codec, which results in a total of 24 evaluations per test subject. The two anchors are further evaluated by the test subjects without their knowledge in order to assess the consistency of their judgement, such that a grand total of 30 video clips is evaluated by each test subject. The latter are shown in random order to the test subjects.

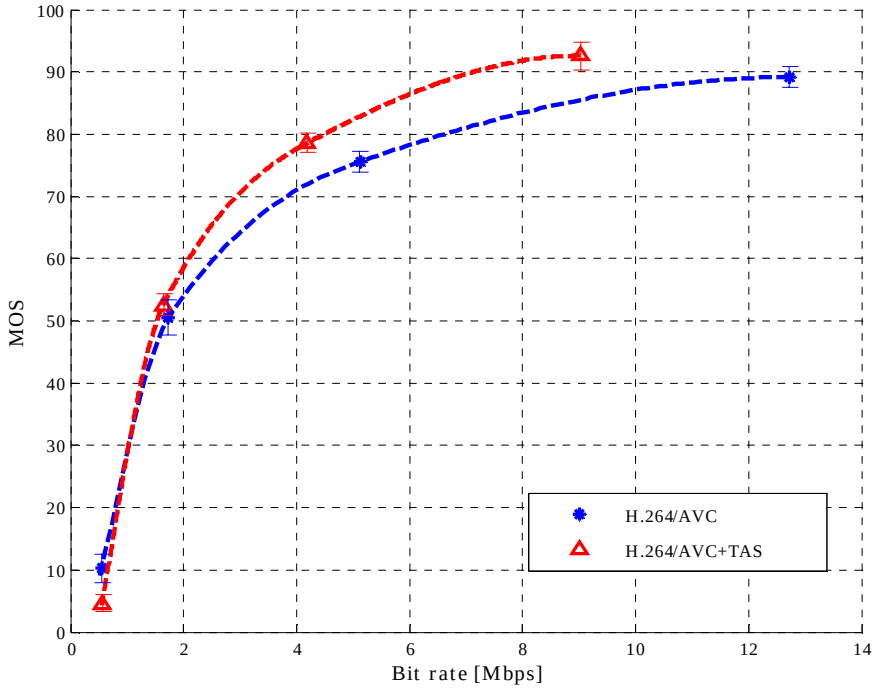


Fig. 103. –Rate-quality curves for "Flower Garden" (genuine H.264/AVC vs. H.264/AVC with texture analysis and synthesis)

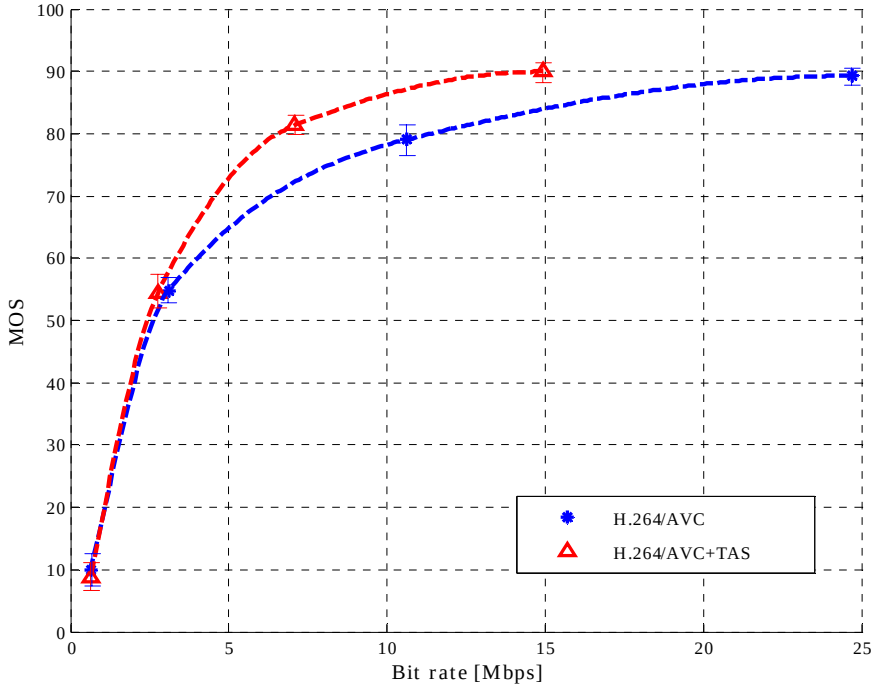


Fig. 104 – Rate-quality curves for "Concrete" (genuine H.264/AVC vs. H.264/AVC with texture analysis and synthesis)

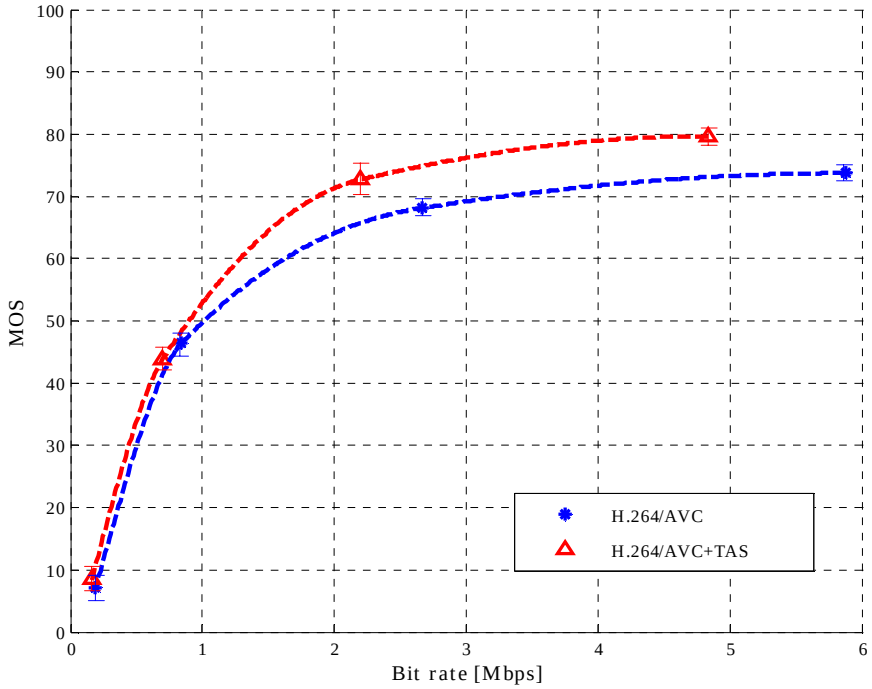


Fig. 105 – Rate-quality curves for "Sea" (genuine H.264/AVC vs. H.264/AVC with texture analysis and synthesis)

Fig. 103, Fig. 104, and Fig. 105 show rate-quality curves for the video sequences "Flower Garden", "Concrete", and "Sea" respectively. 95% confidence intervals are also given in Fig. 103, Fig. 104, and Fig. 105 at each QP value. It can be seen that, for all test sequences, the (red) curve corresponding to the proposed method has better characteristics than the blue curve (H.264/MPEG4-AVC). For QP 32 and QP 40, both codecs give similar performance. Such that the measured differences between the codecs can not be considered as statistically relevant. However, at higher bit rates, statistically relevant deviations are observed between the curves. For "Flower Garden", the subjective bit rate gain achieved by the proposed codec can be given as 40% at QP 16 and 24% at QP 24. For "Concrete", the subjective bit rate gain is determined as 41% at QP 16 and 39% at QP 24. Finally, for "Sea", the subjective bit rate gain is given as 17.6% at QP 16 and 18.5% at QP 24.

10.3.2 Cross-Resolution Properties

An important issue to be discussed refers to cross-resolution bit rate gain improvements. A corollary matter is: How does the side information's bit rate load correlate with video resolution growth? Fig. 106 and Fig. 107 provide insight into cross-resolution properties. The aforesaid figures have been generated by determining the bit rate difference between an H.264/MPEG4-AVC codec without the proposed approach and a corresponding codec with the method under test at fixed coding QP levels as given in Sec. 10.2. This choice has been made here to circumvent time consuming subjective experiments. Two video sequences, "Flower Garden" (cp. Fig. 106) and "Concrete" (cp. Fig. 107), are selected to illustrate the proposed codec behavior. It can be seen in both figures that bit rate gains increase with the bit rate, where the ascending slope is particularly steep at CIF resolution. The general gain increase is due to the fact that, for a given resolution, as long as the proposed method is more effective than genuine H.264/MPEG4-AVC, the side information's bit costs are fixed and independent of the coding QP. That is, the proportion of the side information in the overall bit rate decreases with increasing bit rate (or decreasing QP). It can further be seen that bit rate savings of up to 45% are measured for the "Concrete" sequence at CIF resolution, while 29% are measured at the same resolution for the "Flower Garden" video. For the latter sequence, only slightly improved gains or small losses are observed, when corresponding QP points are pairwise compared across the CIF and SD resolutions ($[-0.3\%\ \ 2.9\%]$). The gain margin increases if SD and 720p are compared in the same way ($[0.2\%\ \ 6.6\%]$). Comparing CIF and 720p resolution now gives a gain margin of $[0.8\%\ \ 9.5\%]$. The respective gain margins obtained for "Concrete" are $[-5.8\%\ \ 4.0\%]$, $[0.25\%\ \ 9.6\%]$, and $[-3.0\%\ \ 13.6\%]$. Note that negative gains represent cross-resolution bit rate losses. These indicate that resolution augmentation does not guarantee higher bit rate savings. The reasons for this codec behavior will be examined in the following sections. Further results regarding bit rate gains at fixed coding QP values can be found in Appendix C.

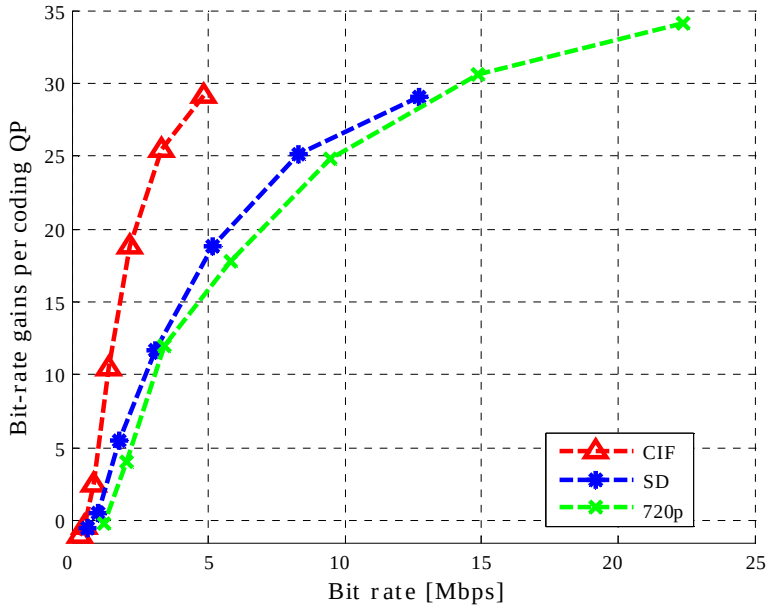


Fig. 106 – Bit rate gains at fixed coding QP values for the “Flower Garden” sequence

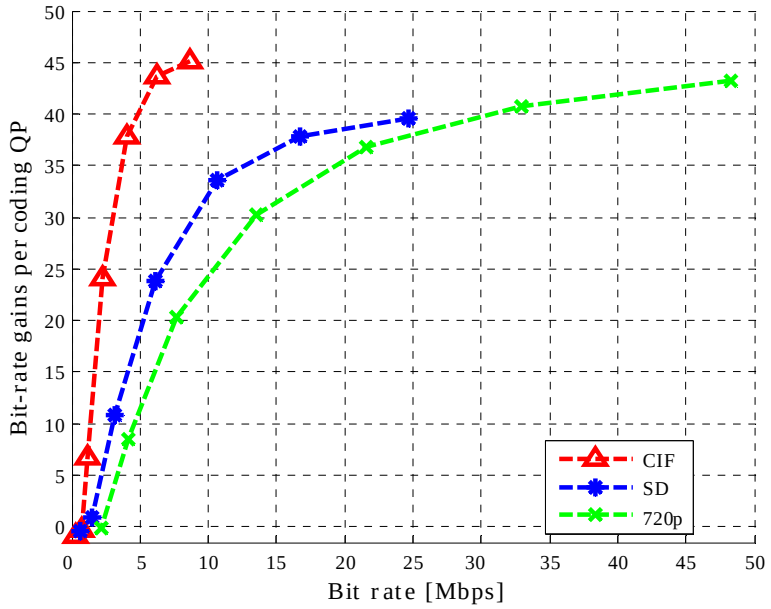


Fig. 107 – Bit rate gains at fixed coding QP values for the “Concrete” sequence

10.3.3 Side Information Properties

The observed cross-resolution bit rate gains or even losses raise some questions about cross-resolution side information behavior. Fig. 108 and Fig. 109 depict the proportion of the content-based side information generated by the proposed approach in the overall bit rate for the "Flower Garden" and the "Concrete" video sequences respectively. CIF, SD, and 720p resolutions are thereby considered. For "Concrete", it can be seen that the side information's contribution to the global bit-rate is typically marginal (<1%) given lower QP values or higher bit rates (cp. Fig. 109, QP 16 and QP 24). The contribution increases to up to 2.3% at QP 32 before dropping to a maximum of 0.5% as from QP 40. It can be noticed that the side information's proportion increases between QP 16 and 32. This relates to the fact that it is almost independent of QP in this interval and for this specific sequence. However does the side information burden gradually increase with the QP value as the overall bit rate decreases. Once the generated side information becomes larger than MSE-related information, the reference codec is selected to operate H.264/MPEG4-AVC coding (cp. Fig. 109, QP 40). Note, however, that the side information is non-zero, when reference coding is conducted, as analysis-synthesis-related signaling is still operated, where empty masks (no detail-irrelevant textures to be coded) are transmitted. Similar observations can be made for "Flower Garden", where at CIF and partly at SD resolution reference coding is selected as from QP 32 already. For that, it can be seen in Fig. 106 and Fig. 107, that, for a given video resolution, there are bit rate losses of up to 0.85% for "Flower Garden" and 1% for "Concrete" at low bit rates compared to the genuine codec. It is worthwhile noting that, for "Flower Garden", the side information's share in the overall bit rate is inverse proportional to the video resolution. This at least applies for higher bit rates. However, this nice relationship between side information and resolution does not hold for the "Concrete" test sequence.

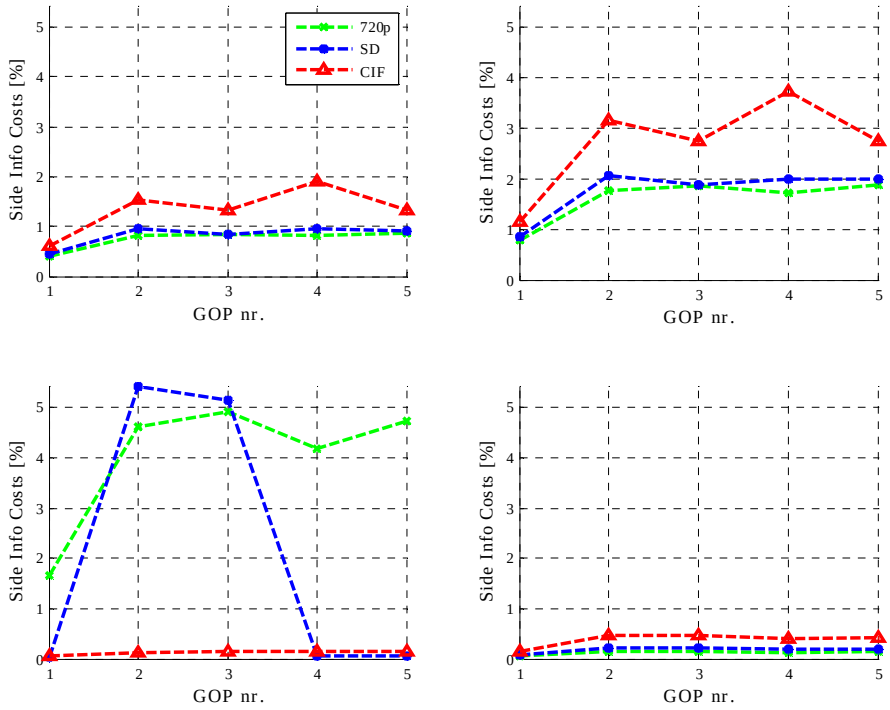


Fig. 108 – Content-based side information costs relatively to overall costs ("Flower Garden"). QP 16 (top left), QP 24 (top right), QP 32 (bottom left), QP 40 (bottom right).

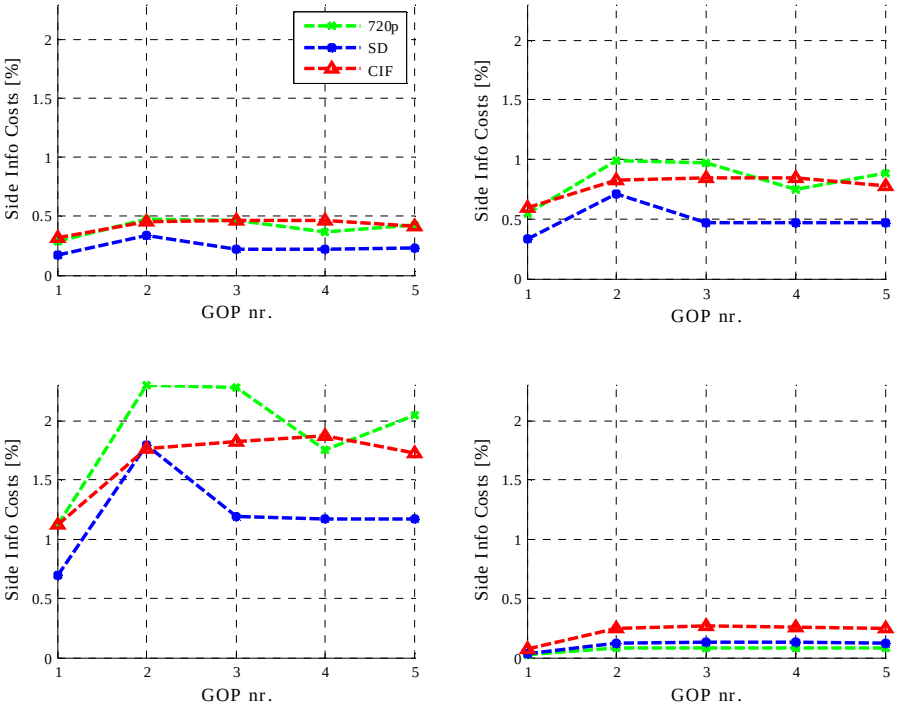


Fig. 109 – Content-based side information costs relatively to overall costs ("Concrete"). QP 16 (top left), QP 24 (top right), QP 32 (bottom left), QP 40 (bottom right).

Fig. 108 and Fig. 109 show that the side information transmitted to the decoder varies from resolution to resolution. Beyond coding cost issues, cross-resolution deviations in the side information may have other causes. For instance can they be explained by different texture analysis results at each resolution. This is exemplarily depicted in Fig. 110 and Fig. 113, where the cross-resolution segmentation masks are shown for “Flower Garden” and “Concrete”. It can be seen that, for the former test sequence, the synthesized area (non-black samples in the masks) gradually increases from resolution to resolution (14.3% from CIF to SD, 12.1% from SD to 720p, and 26.4% from CIF to 720p). This is further shown in Fig. 111. However, the cross-resolution bit rate gains (overall maximum of 9.5% from CIF to 720p) do not keep up with the observed synthetic area growth figures. A better insight into this phenomenon can be achieved by considering macroblock bit costs in the genuine H.264/MPEG4-AVC codec. Fig. 112 depicts masks showing such costs, where the highest cross-resolution macroblock cost in the given video sequence is mapped to white and the lowest cost is mapped to black. The synthetic segments are marked by a red boundary. It can be seen that, comparing CIF and SD resolutions, synthetic area increase occurs both in low-cost as well as in high-cost areas, where low-cost-related enlargements dominate by far. For that, slight bit rate gains of up to 2.9% are justified here (cp. Sec. 10.3.2). Comparing SD and 720p resolutions now shows that an important synthetic area growth can be observed in the low-cost area. However, the high-cost macroblocks in the sky texture (branches, cp. Fig. 110) and the windmill are submitted to synthesis at 720p resolution, which is not the case at SD resolution. For that, moderate cross-resolution gains of up to 6.6% are observed. Finally, comparing CIF and 720p resolutions, it shows that large synthetic area increase is given in both low- and high-cost areas, which explains the maximum bit rate gain of 9.5% here.

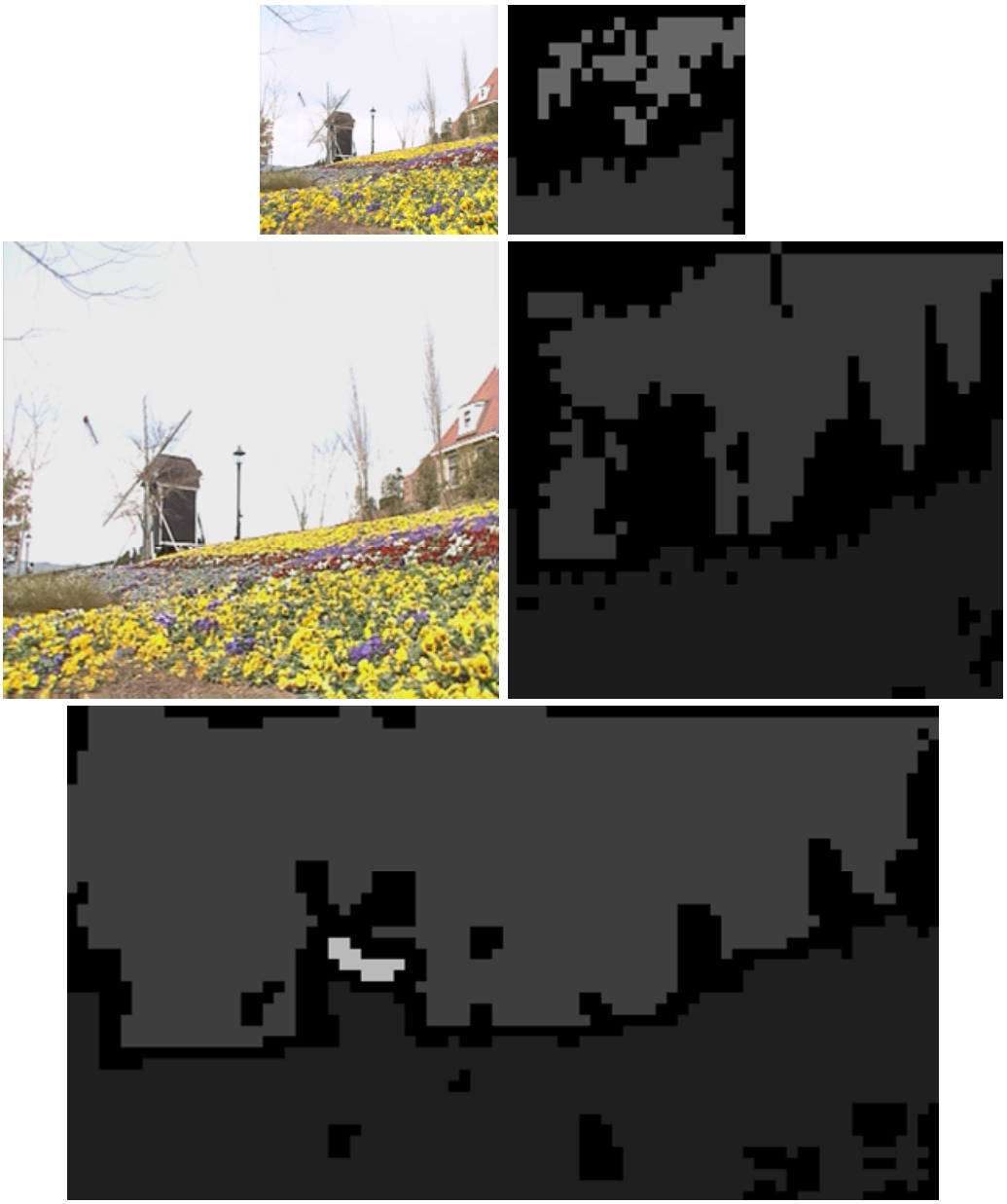


Fig. 110 – Synthetic picture areas in “Flower Garden” at CIF (top row), SD (middle row) and 720p resolutions (bottom). Masks with synthetic areas (non-black segments) shown in right column (first and middle row) and at the bottom. Original pictures shown in left column (first and middle row).

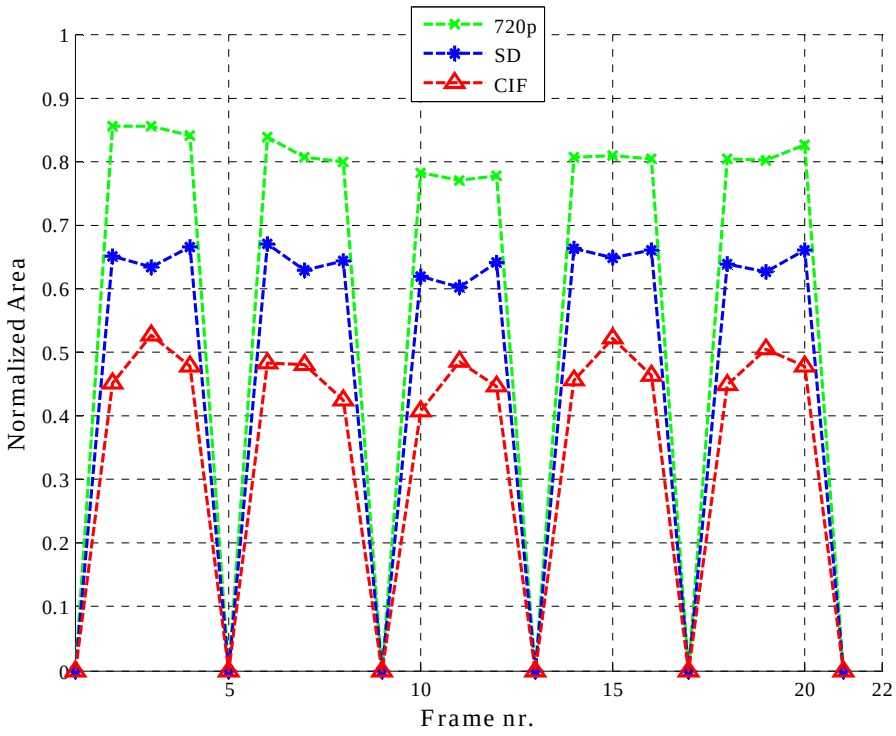


Fig. 111 – Synthesized picture area at different video resolutions (“Flower Garden”)

For “Concrete”, the synthesized area decreases from CIF to SD by 7.3 %, while it increases by 17.3% from SD to 720p and by 10.1% from CIF to 720p (cp. Fig. 114). It can be seen in Fig. 113 that, at SD resolution, TSI post-processing yields a significantly decimated mask at the picture borders compared to the other resolutions. This can be explained by the requirement of a macroblock accurate segmentation mask, covering and uncovering effects (cp. Sec. 5.2.2), as well as mask corrections through local video quality analysis (cp. Sec. 6.4.4). As the macroblocks at the picture border are relatively costly (cp. Fig. 115), cross-resolution bit-rate loss of up to 5.8% is observed at higher bit rates (QP 16-20). Note that, at these bit rates, texture synthesis is conducted in every GOP. Although significant area increase is observed between CIF and 720p, cross-resolution bit rate loss of up to 3% is seen (QP 16-24). This can be explained by the fact that macroblock costs are significantly more important at CIF than at 720p resolutions (cp. Fig. 115).

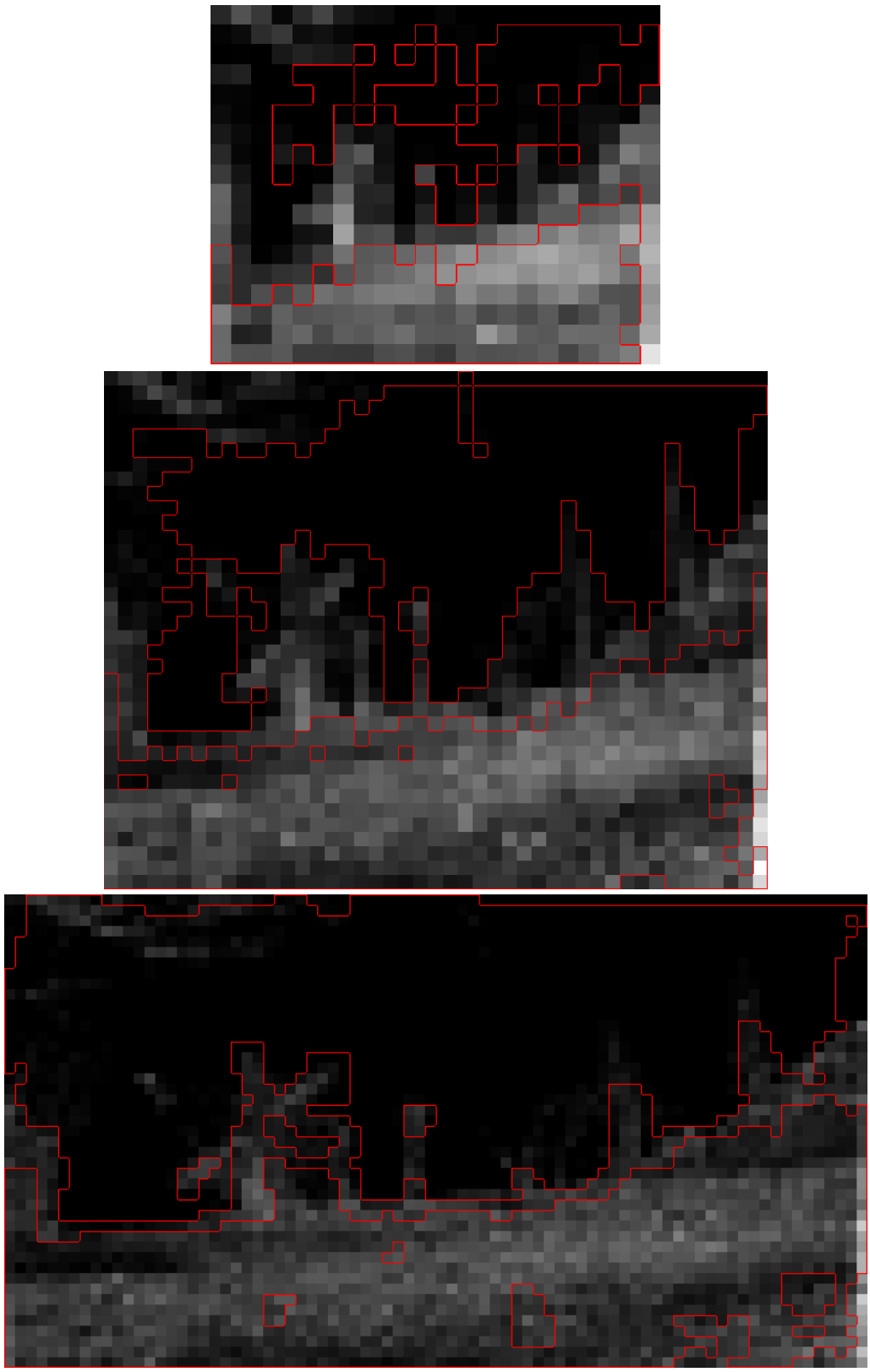


Fig. 112 – Bit costs of synthetic picture areas (delimited by red boundaries) in “Flower Garden” at QP 16 for CIF (top), SD (middle) and 720p resolutions (bottom)

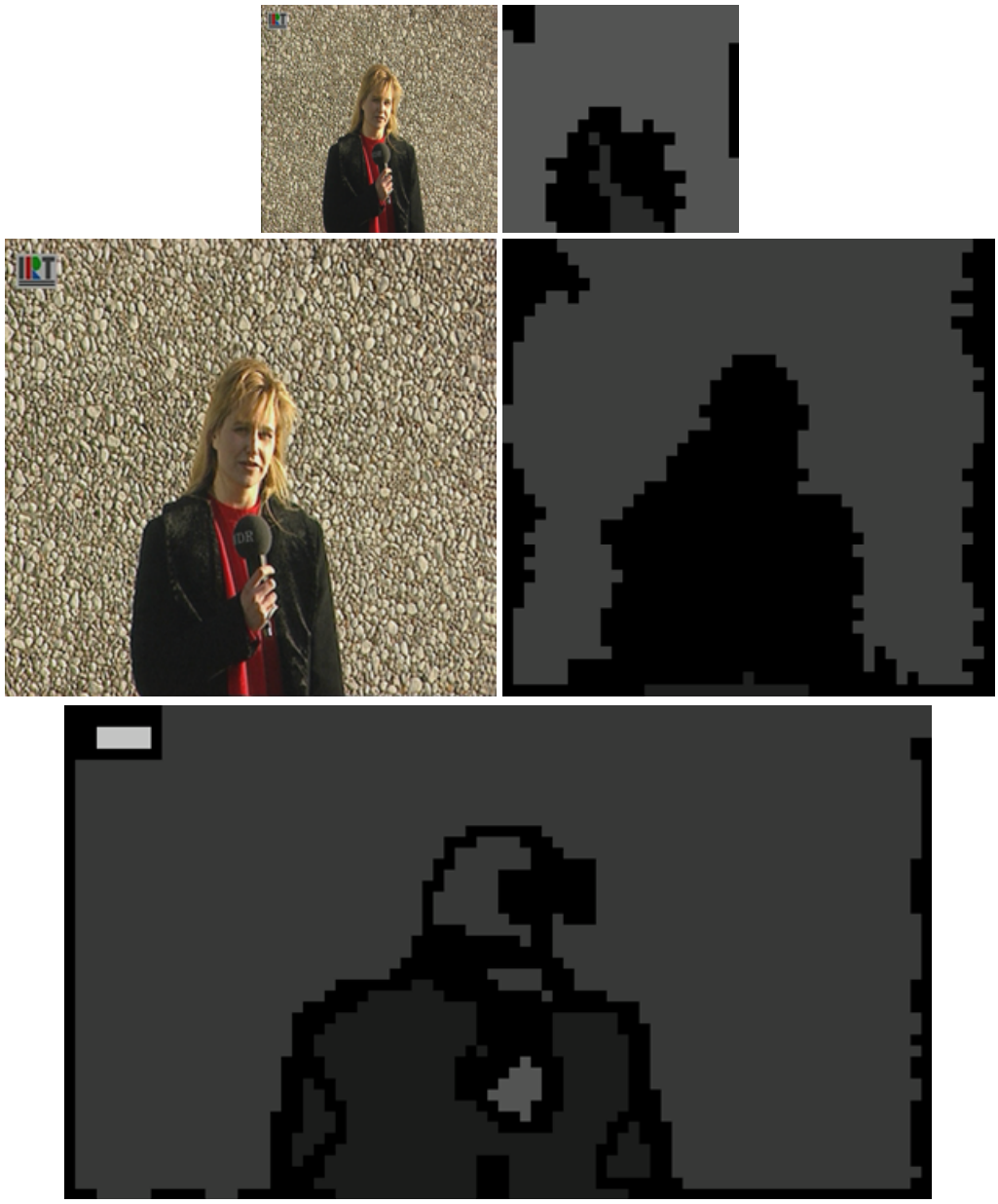


Fig. 113 –Synthetic picture areas in “Concrete” at CIF (top row), SD (middle row) and 720p resolutions (bottom). Masks with synthetic areas (non-black segments) shown in right column (first and middle row) and at the bottom. Original pictures shown in left column (first and middle row).

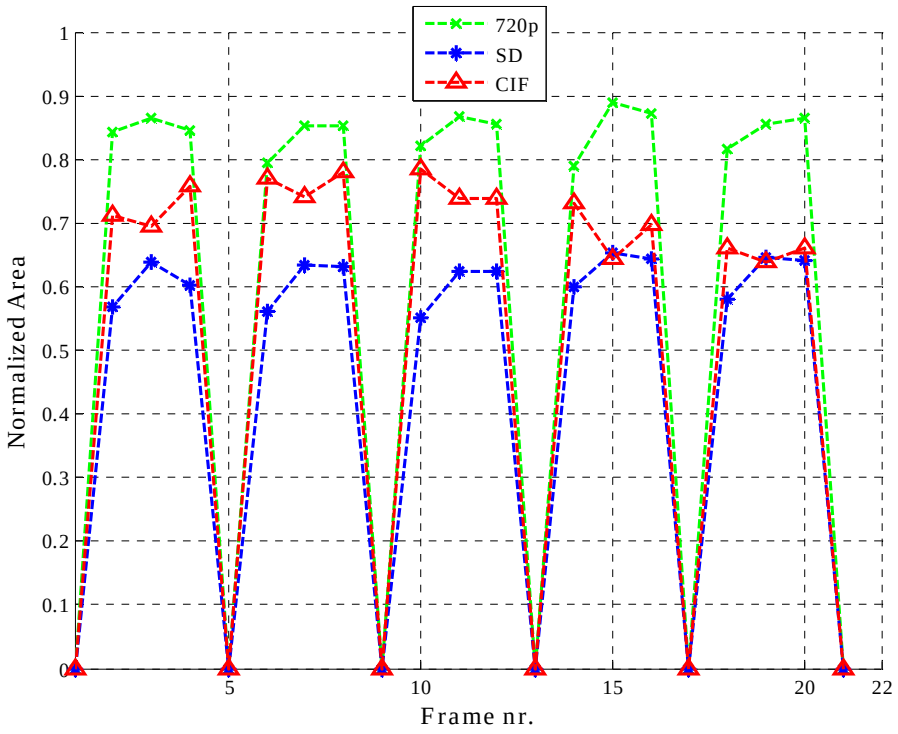


Fig. 114 – Synthesized picture area at different video resolutions (“Concrete”)

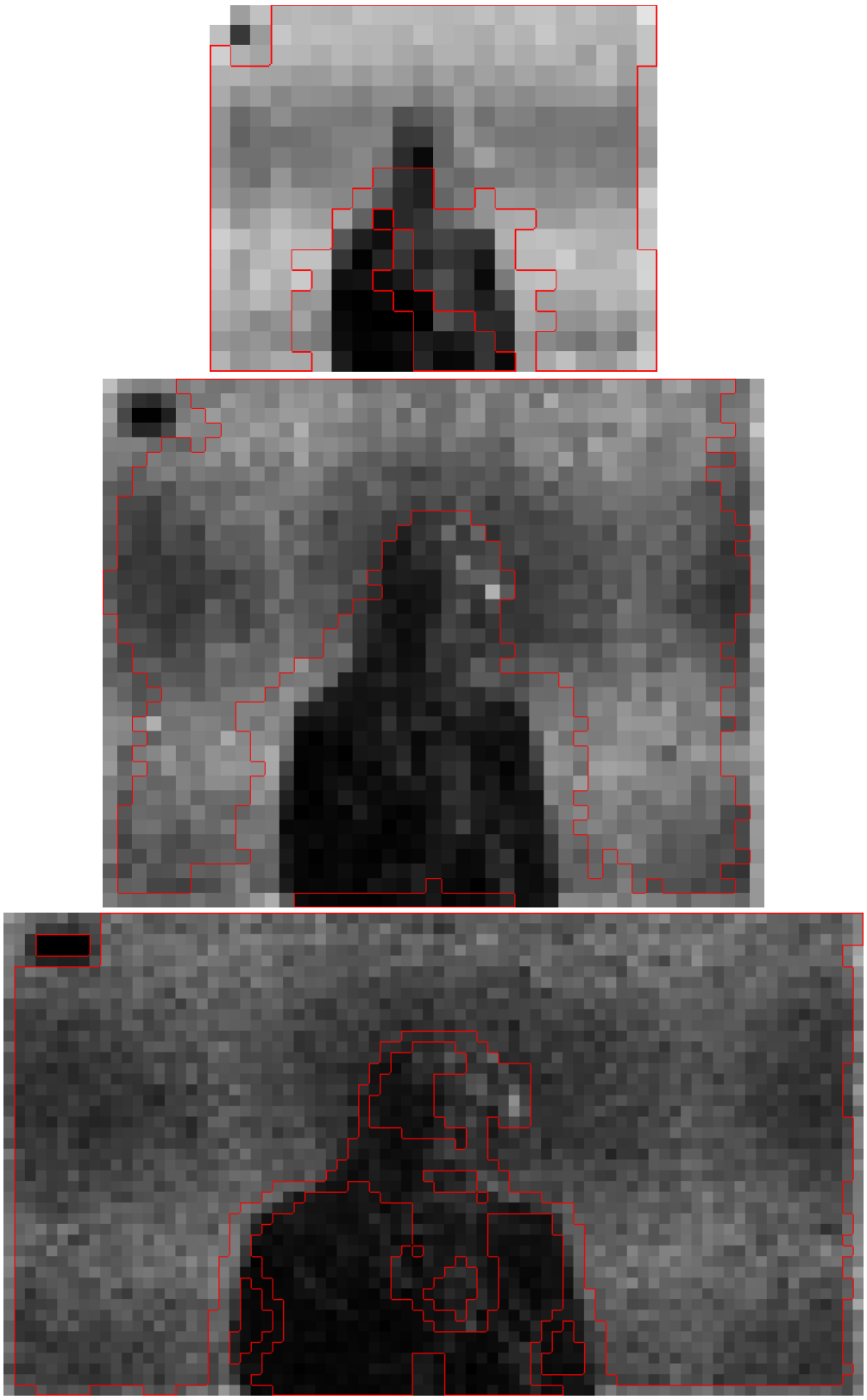


Fig. 115 – Bit costs of synthetic picture areas (delimited by red boundaries) in “Concrete” at QP 16 for CIF (top), SD (middle) and 720p resolutions (bottom)

10.3.4 Modules-Related Considerations

The influence of the local video quality assessment measures (cp. Sec. 6.4.4) on the overall synthesis outcome is shown in Fig. 116 and Fig. 117. The latter depict the relative texture synthesis (TS) area over the pictures of the "Flower Garden" and the "Concrete" sequence respectively and at CIF resolution. It can be seen that the spatio-temporal texture analyzer identifies a relatively large area in each picture as detail-irrelevant (blue curve in Fig. 116 and Fig. 117). The initial mask by the texture analyzer is corrected through local VQA (cp. Sec. 6.4.1 and Sec. 7.3). Reference pictures are not modified by this approach (cp. pictures 1,5,9, … in Fig. 116 and Fig. 117). The B picture masks are, however, significantly corrected to improve synthesis quality. The mean synthetic area is corrected from 54% to 34% for "Flower Garden" and from 62% to 52% for "Concrete".

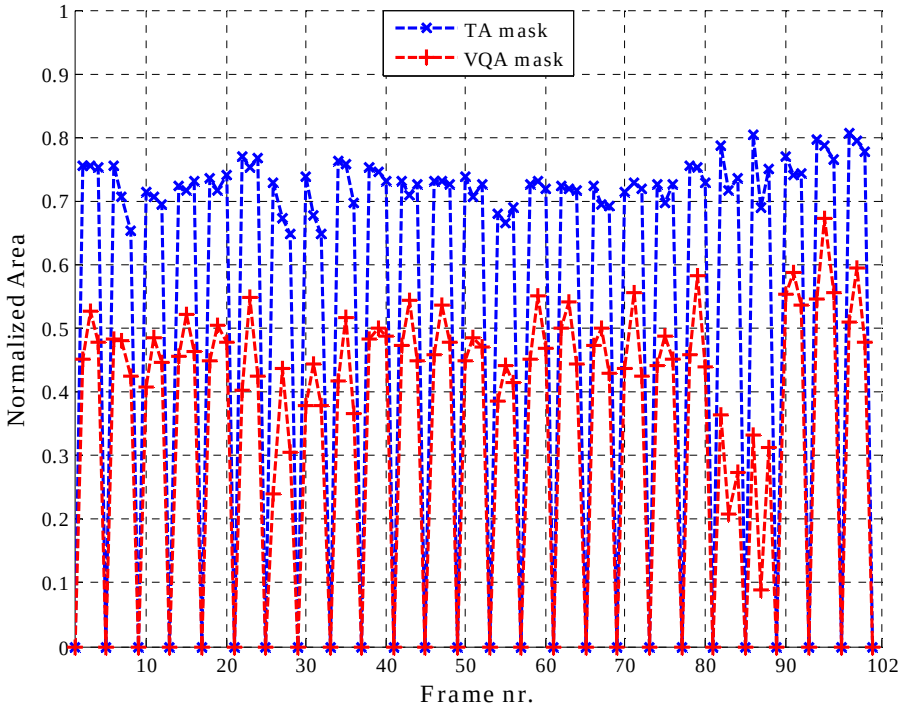


Fig. 116 – Impact of mask correction through local video quality assessment ("Flower Garden")

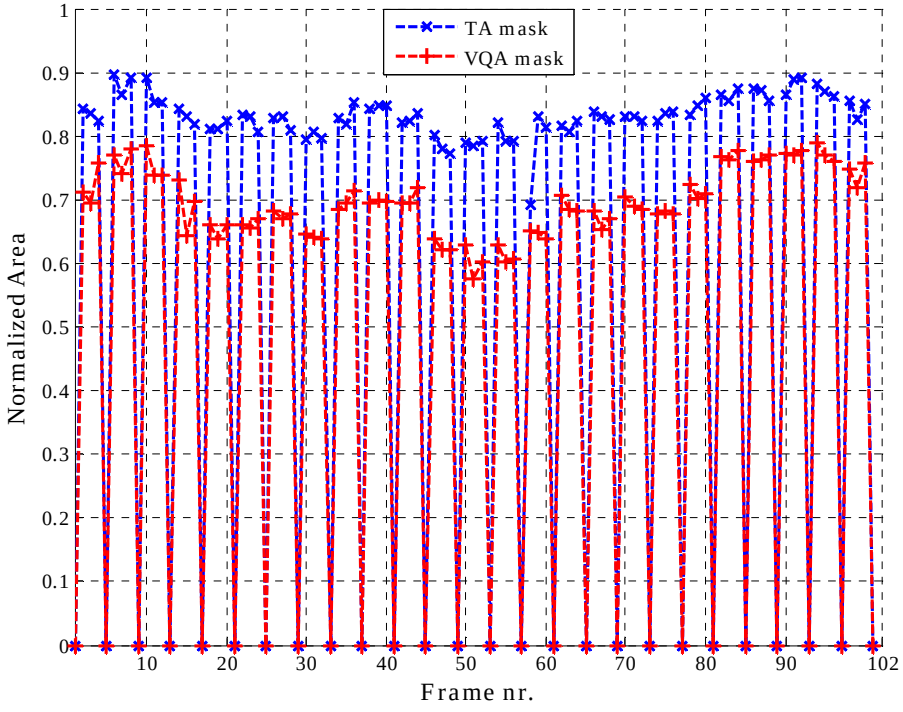


Fig. 117 – Impact of mask correction through local video quality assessment ("Concrete")

Fig. 118 and Fig. 119 show segmentation masks, where the hatched areas correspond to detail-irrelevant textures. The picture on the left hand side corresponds to the mask obtained from the texture analyzer, while the right picture is the mask after correction by the local video quality measure. It can be seen that in both cases the amount of segmentation mistakes is significantly reduced by the local VQM. The effectiveness of the latter is implicitly validated by the subjective experiments conducted in the previous sections.

Fig. 118 – Mask correction through local video quality assessment ("Concrete"). Texture analyzer mask (left), updated mask (right).

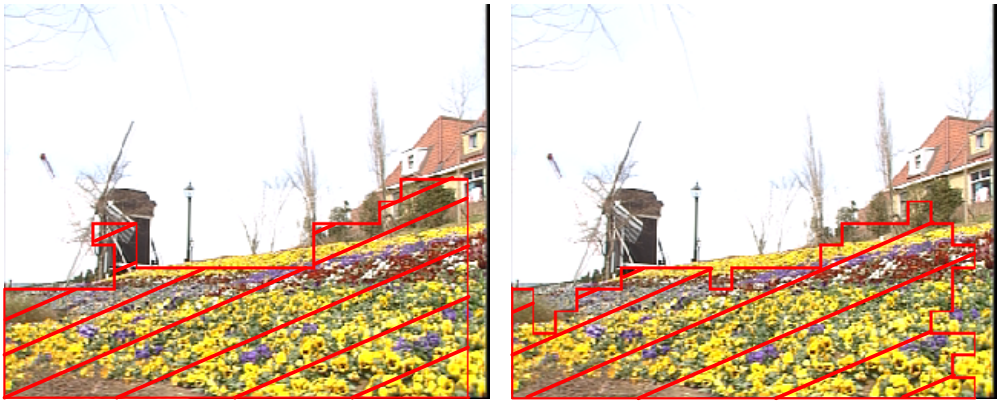


Fig. 119 – Mask correction through local video quality assessment ("Flower Garden"). Texture analyzer mask (left), updated mask (right).

10.3.5 Intrinsic H.264/MPEG4-AVC Modes

Given the properties of the proposed framework, it is now to be verified if the gains observed above (cp. Secs. 10.3.1 and 10.3.2) may be achieved through intrinsic H.264/MPEG4-AVC features. For that, it is recalled that the new approach can be seen as PSNR deterioration of specific picture segments without negative incidence on the subjective signal quality. Hence, simulation of the proposed framework with genuine H.264/MPEG4-AVC means is done by coding regions to be synthesized at an artificially high coding QP such that the target bit rate given by the proposed framework is matched. The corresponding codec will be referred to as the manipulated codec in the following.

Fig. 120 – "Concrete" test image generated with the manipulated H.264/MPEG4-AVC codec (top) and with the proposed video codec (bottom)

All video sequences evaluated in the overall framework are considered in these experiments. Fig. 120 shows results achieved by the two codecs under test for "Concrete" at CIF resolution. The latter are compared at a bit rate of 1115 kbps, where QP is set to 32. The manipulated codec is forced to encode regions to be synthesized at a QP of 40. It can be seen that the manipulated codec yields noticeable spatial artifacts. The temporal inconsistencies are even worse.

The same is observed for the "Sea" test sequence at CIF resolution as shown in Fig. 121 (700 kbps, QP=32, manipulated QP=35). The artifacts of the manipulated codec relate to blurred areas, while more details are preserved in the decoded sequence of the proposed framework.

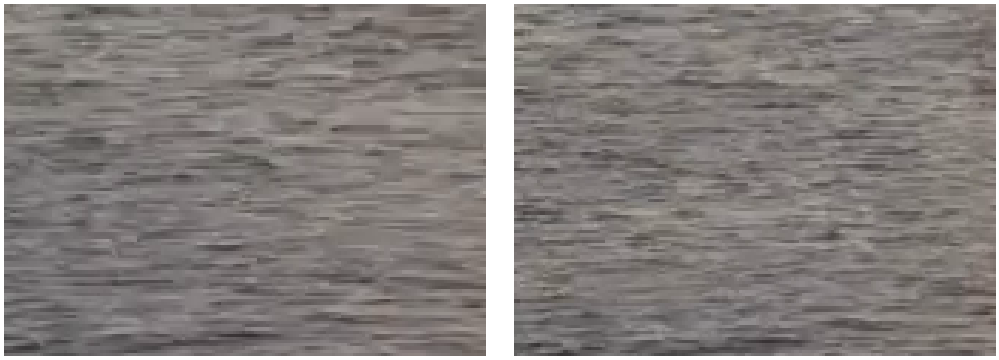


Fig. 121 – "Sea" test region generated with the manipulated H.264/MPEG4-AVC codec (left) and with the proposed video codec (right)

However, for "Flower Garden", no significant deterioration of the subjective video quality was noticed. This is due to the translational camera motion in this test sequence that is obviously well predictable by the cheap SKIP mode.

11 Conclusions

An automatic, content-based approach for improved H.264/MPEG4-AVC video coding was presented in this thesis. Consistent quality of the decoded video signal is ensured by operating closed-loop analysis-synthesis with incorporated objective measurement of subjective quality. The fundamental hypothesis of the proposed approach is that textures in a video sequence can be classified into two classes: Perceptually relevant and perceptually irrelevant. Only the latter textures are described at reduced bit rate via meta data that are transmitted as side information to the decoder, where they are used for texture regeneration. Perceptually relevant textures are H.264/MPEG4-AVC coded. Target texture detection is done implicitly via the implemented rate-distortion decision module. That is, no explicit classification of textures into the above-mentioned classes is conducted. The experiments show that the proposed algorithm yields bit rate savings of up to 41% compared to a standard conforming H.264/MPEG4-AVC video codec.

Although the proposed video coding approach outperforms H.264/MPEG4-AVC, it appears that major gains are achieved at bit rates that lie beyond the usual operating points of known transmission systems. For instance, the 41% bit rate savings observed above are obtained at a bit rate of approximately 25 Mbps, whereas the Digital Video Broadcasting Terrestrial (DVB-T) standard provides only 3 to 5 Mbps per TV channel. Hence, future work will focus on enhancing the greed of the presented framework. Two major, synthesis-related paths are expected to yield significant bit rate reduction at practical operating points. The first path relates to improved handling of covering and uncovering effects. For rigid textures, synthesis results will be improved through enhanced intra-synthesis algorithms. Non-rigid textures will be enhanced by operating recursive synthesis given significant camera motion. The second path relates to performance improvement of the proposed texture synthesizers for longish but costly regions, as texture transitions or foreground objects. In fact, it is believed that the degree of dominant-structure-awareness of future synthesis tools will play a key role in the subjective performance of these. This is justified by the fact

that object contours as well as dominant edges in textures should ideally be preserved by the synthesis method.

Texture analysis performance will be increased, where particular attention will be paid to the over-segmentation tendency of the proposed method. The absolute error rates will also be reduced. For that, long-term criteria for segment trajectory observation will be integrated into the spatio-temporal analyzer for improved merger decisions. Automatic weighting of spatial and temporal hints appears to be a further important approach to avoid a priori prioritization of any of these features.

The proposed RD optimization approach is viable but does not provide a joint optimization of rate and distortion variables. Problems relating to the fact that spatio-temporal dependencies exist between macroblocks in the proposed codec will be tackled in future work. In texture synthesis, such dependencies arise from motion compensation (translational model for non-rigid texture synthesis and perspective model for rigid texture synthesis). Finally, the granularity of the rate-distortion optimization process will be refined from group-of-pictures to texture region accuracy.

A. Results of Spatio-Temporal Texture Analysis

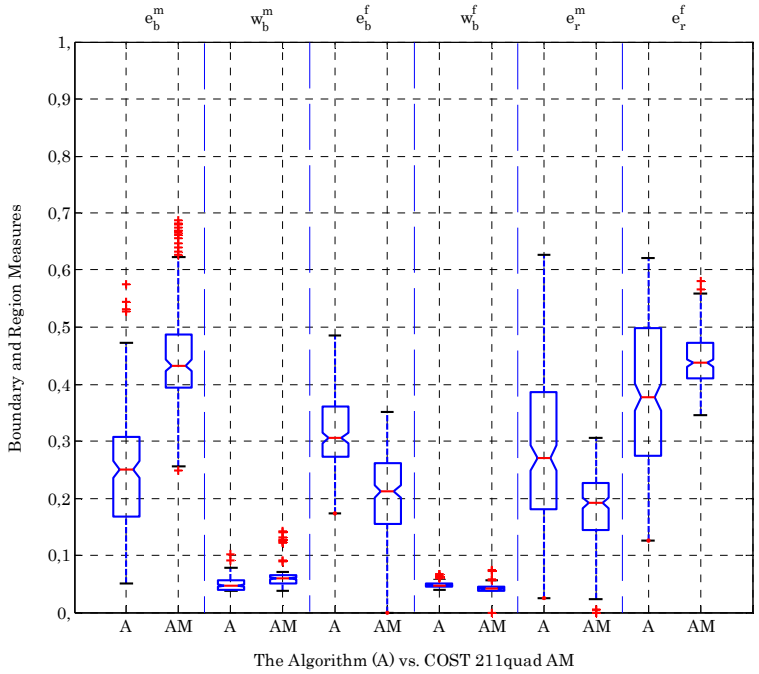


Fig. 122 – Boxplots of Huang and Dom's measures for the "Canoe" sequence

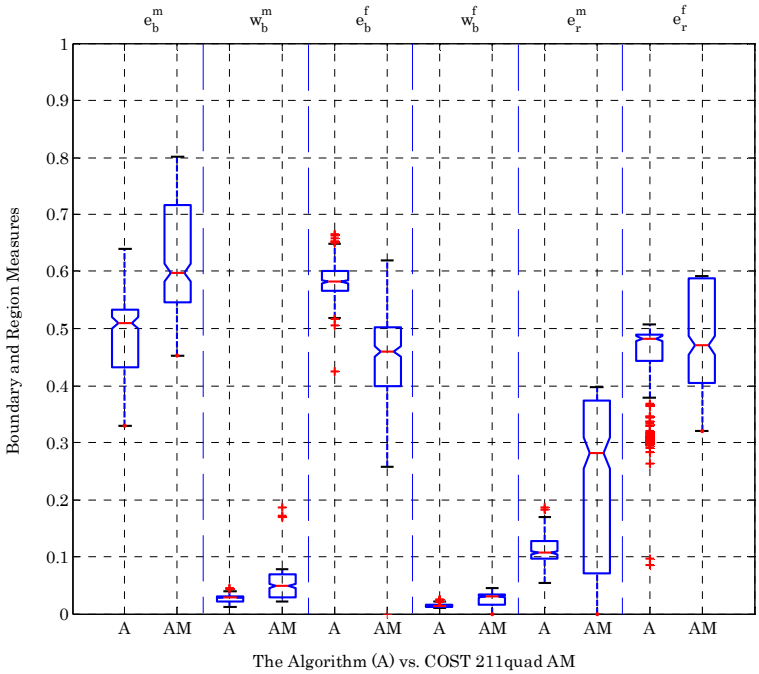


Fig. 123 – Boxplots of Huang and Dom's measures for the "Coast Guard" sequence

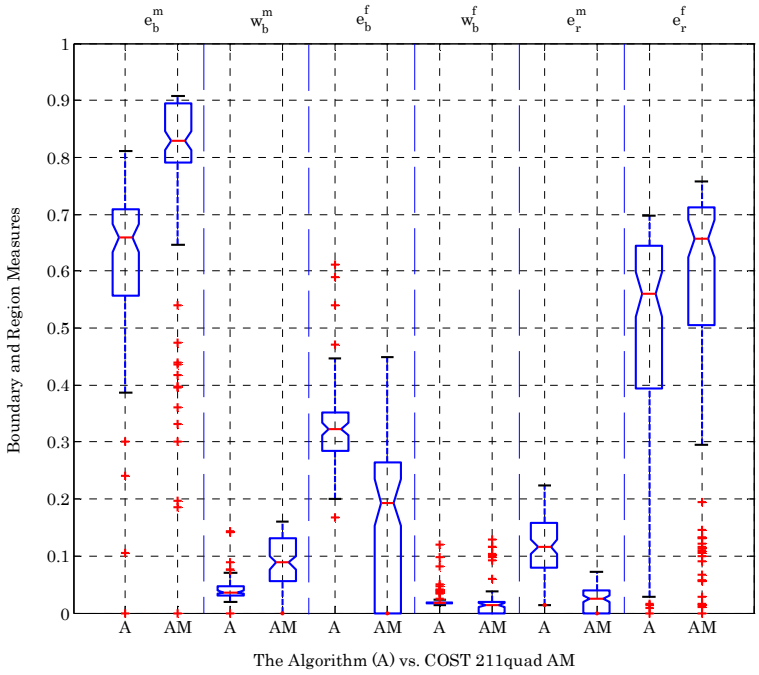


Fig. 124 – Boxplots of Huang and Dom's measures for the "Football" sequence

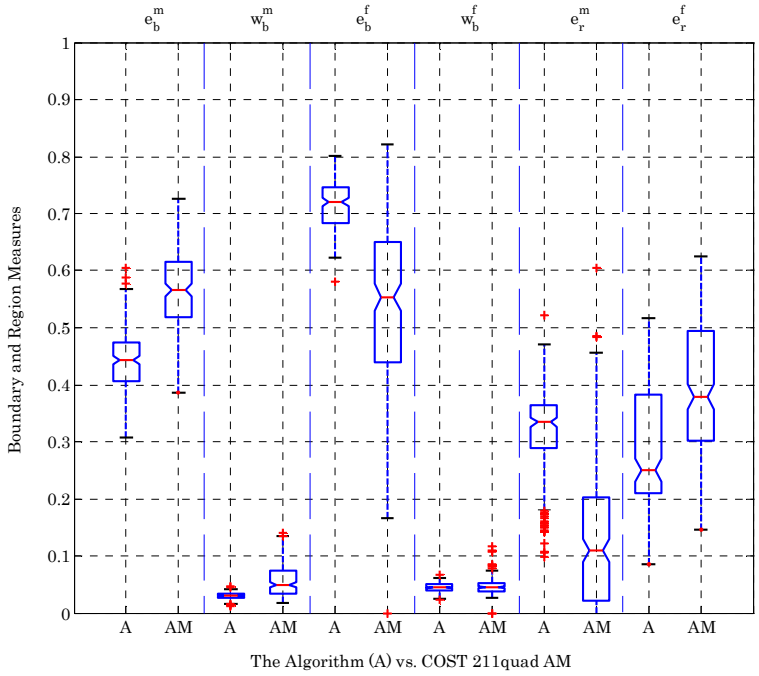


Fig. 125 – Boxplots of Huang and Dom's measures for the "Foreman" sequence

B. Video Quality Assessment

B.1 Configuration of Ong et al.'s VQM [184],[185]

Parameters	Setting
f_{BF}	0.25
f_{RF}	0.25
γ_1, γ_2	10^{-10}
s	0
C^{is}	1
f_s	1
f_r	variable
$T_{m,o}$, $T_{m,1}$	1
$T_{m,2}$, L_m	0
$T_{m,3}$	0.1
z_1, z_2	e
$\alpha_1, \alpha_2, \alpha_3$	1
α_4	0
$T_{l,o}$, $T_{l,1}$	1
$T_{l,2}$, L_l	0
r	0.1
ν_1, ν_2	e

B.2 VQM Complexity

PSNR

The complexity of PSNR can be determined as

$$Pr^1_{PSNR} = 6 + MN;\ Add^1_{PSNR} = 2MN \tag{134}$$

where Pr^1_{PSNR} and Add^1_{PSNR} represent the number of products/divisions and the number of additions/subtractions respectively. These complexity estimations do however not include the logarithmic function. A Taylor series representation of the latter can be given as

$$\ln(x) = \sum_{n=0}^{\infty}(-1)^n \frac{(x-1)^{n+1}}{n+1} \approx \sum_{n=0}^{N_{Taylor}}(-1)^n \frac{(x-1)^{n+1}}{n+1}\ . \tag{135}$$

Hence, the complexity of $\log_{10}(x)$ can be obtained as

$$Pr^2_{PSNR} \approx 3N_{Taylor}(N_{Taylor}+1);\ Add^2_{PSNR} = 2(2N_{Taylor}+1) \tag{136}$$

where the condensed form of the arithmetic series

$$1+2+3+...+N_{Taylor}=\frac{N_{Taylor}(N_{Taylor}+1)}{2} \tag{137}$$

has been used to simplify the expressions of Pr^2_{PSNR} and Add^2_{PSNR}. It is assumed here that the Taylor series above is interrupted after N_{Taylor} terms without significant loss of accuracy. It can be further supposed that $MN >> 3N_{Taylor}(N_{Taylor}+1)$ and $MN >> 2(2N_{Taylor}+1)$. The (over-) estimated overall PSNR complexity can now be given as

$$Pr_{PSNR}=Pr^1_{PSNR}+Pr^2_{PSNR}=6+2MN;\ Add_{PSNR}=Add^1_{PSNR}+Add^2_{PSNR}=3MN \tag{138}$$

where Pr^2_{PSNR} and Add^2_{PSNR} have been set to MN each in order to simplify the expression of the overall PSNR complexity.

Proposed VQM

The complexity of the proposed spatial VQM can be determined as

$$Pr^1_{VQM} \approx \frac{78}{\kappa}MN+3=4.87MN+3;\ Add^1_{VQM} \approx \frac{13}{\kappa}MN+2=1.625MN+2\ . \tag{139}$$

Notice that boundary issues in filtering operations are ignored here. The exponential function in the proposed VQM has not been considered so far. A Taylor series representation of the former can be given as

$$e^x=\sum_{n=0}^{\infty}\frac{x^n}{n!} \approx \sum_{n=0}^{N'_{Taylor}}\frac{x^n}{n!}\ . \tag{140}$$

Hence, the complexity of e^x can be obtained as

$$Pr^2_{VQM} \approx N'_{Taylor}(N'_{Taylor}+2)\ ;\ Add^2_{VQM}=N'_{Taylor} \tag{141}$$

where (137) has been used to condense the expressions of Pr^2_{VQM} and Add^2_{VQM}. It is assumed again that the Taylor series above is interrupted after N'_{Taylor} terms without significant loss of accuracy. It can be further assumed that $MN >> N'_{Taylor}$

and $MN >> N'_{Taylor}\,(N'_{Taylor}+2)$. The (over-) estimated overall VQM complexity can now be given as

$$Pr_{VQM} = Pr^1_{VQM} + Pr^2_{VQM} \approx 5.87MN + 3;\; Add_{VQM} = Add^1_{VQM} + Add^2_{VQM} \approx 2.625MN + 2 \quad (142)$$

where Pr^2_{VQM} and Add^2_{VQM} have been approximated by MN each.

Notice that a detailed derivation of the complexities of PSNR and the proposed spatial VQM is given in appendix B.2.

C. Performance of Overall Framework

Bit rate gains obtained using an H.264/MPEG4-AVC codec with the proposed approach, compared to a corresponding codec without the new method, are depicted in Fig. 126 and Fig. 127. The "Sea", "Life Belt", and "Synchronized Swimming" sequences belong to the test sets described in Tab. 5 and Tab. 9, while the "News", "Cartoon", and "Color Movie" clips belong to the TV test data described in Tab. 7.

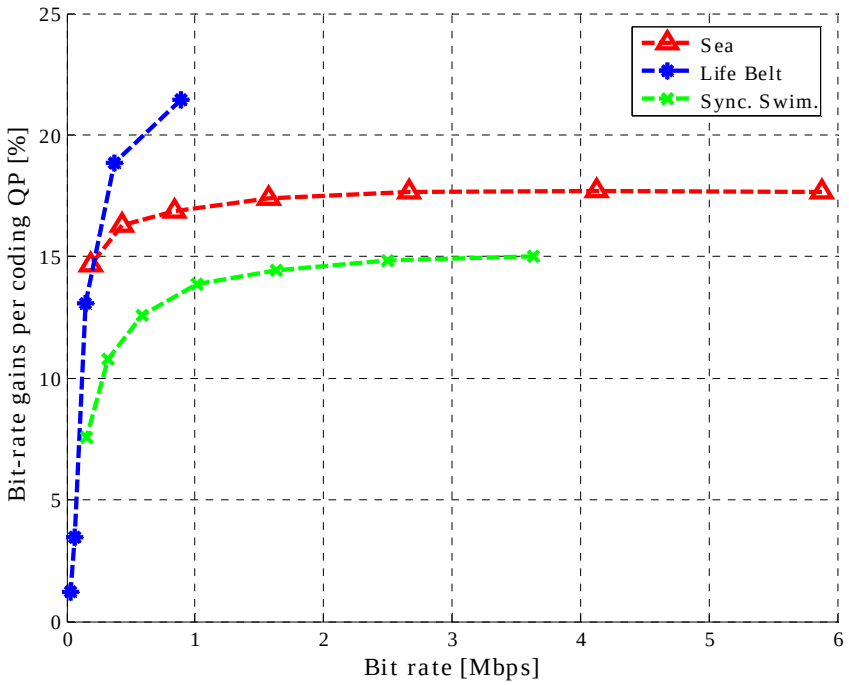


Fig. 126 – Bit rate gains at fixed coding QP values for the "Sea", "Life Belt", and "Synchronized Swimming" sequences

Fig. 126 depicts additional results for video sequences with non-rigid textures. It can be seen that bit rate savings of up to 22% are achieved here. It should, however, be noted that some artifacts are visible in the "Synchronized Swimming" clip due to reflections in the water that can not be handled by the corresponding synthesizer. Detection of these annoyances works only partially for this sequence.

The results shown in Fig. 127 refer to sequences with rigid textures. Limited gains are achieved here due to the low motion and texture complexity of the video clips. Negative bit rates indicate that the proposed codec yields higher bit rate than the reference codec. This is due to the side information load, as explained in Sec. 10.3.3.

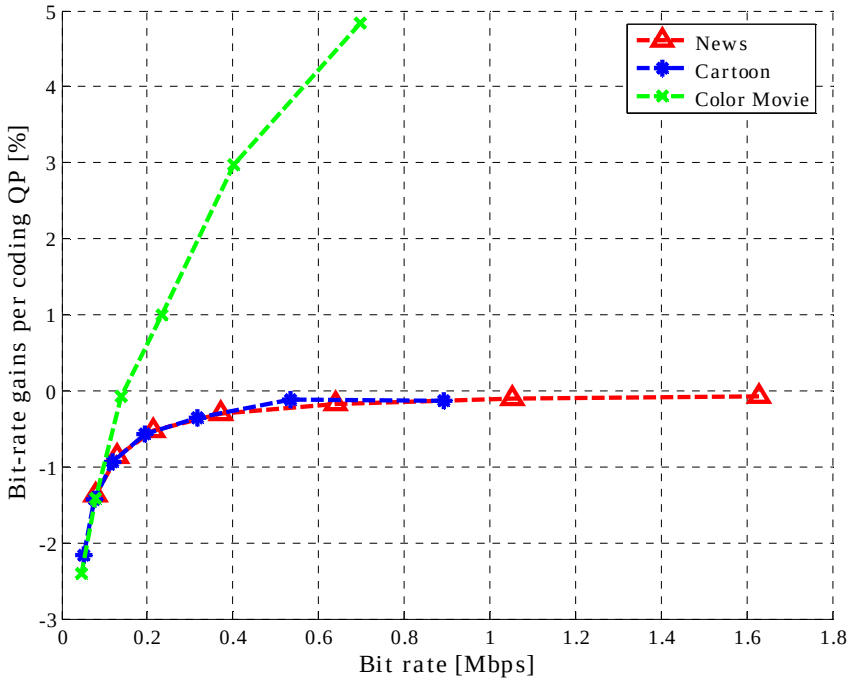


Fig. 127 – Bit rate gains at fixed coding QP values for the "News", "Cartoon", and "Color Movie" sequences

Bibliography

[1] F. Pereira and T. Ebrahimi, "The MPEG-4 Book", ISBN 0-130-61621-4, Prentice Hall, New Jersey, 2002.

[2] B. S. Manjunath, P. Salembier, and T. Sikora, "Introduction to MPEG-7", ISBN 0-471-48678-7, Wiley, Sussex, England, 2003.

[3] ITU-T Rec. H.264 and ISO/IEC 14496-10 (MPEG4-AVC), "Advanced Video Coding for Generic Audiovisual Services", v1, May 2003; v2, Jan. 2004; v3 (with FRExt), Sept. 2004; v4, July 2005.

[4] J.-R. Ohm, "Multimedia Communication Technology", ISBN 3-540-01249-4, Springer, Berlin Heidelberg New York, 2004.

[5] T. Sikora, "Trends and Perspectives in Image and Video Coding", Proc. of the IEEE, Vol. 93, No. 1, p. 6-17,January 2005.

[6] P. Eisert, "Very Low Bit-Rate Video Coding using 3-D Models", PhD Thesis, ISBN 3-8265-8308-6, Shaker Verlag, Aachen, Germany, 2000.

[7] M. Kunt, A. Ikonomopoulos, and M. Kocher, "Second Generation Image Coding Techniques", Proc. of the IEEE, Vol. 73, No. 4, p. 549-575, April 1985.

[8] L. Torres and M. Kunt, "Video Coding: The Second Generation Approach", Kluwer Academic Publishers, 1996.

[9] P. Salembier, L. Torres, F. Meyer, and C. Gu, "Region-based Video Coding using Mathematical Morphology", Proc. of the IEEE , Vol. 83, No. 6, p. 843–857, June 1995.

[10] P. Salembier and F. Marqués, "Region-based Representations of Image and Video: Segmentation Tools for Multimedia Services", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 9, No. 8, p. 1147-1169, December 1999.

[11] T. Chen, C. T. Swain, and B. G. Haskell, "Coding of Subregions for Content-based Scalable Video", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 7, No. 1, p. 256-260, February 1997.

[12] H. G. Musmann, M. Hötter, and J. Ostermann, „Object-oriented Analysis-Synthesis Coding of Moving Images", EURASIP Signal Processing, Vol. 1, No. 2, p. 117-138, October 1989.

[13] E. H. Adelson, "Layered Representation for Image Coding", Technical Report 181, The MIT Media Lab, 1991.

[14] P. Willemin, T. Reed, and M. Kunt, "Image Sequence Coding by Split and Merge", IEEE Transactions on Communications, Vol. 39, No. 12, p. 1845-1855, December 1991.

[15] P. Bouthemy and E. François, "Motion Segmentation and Qualitative Dynamic Scene Analysis from an Image Sequence", International Journal of Computer Vision, Vol. 10, No. 2, p. 157-182, 1993.

[16] J. Y. A. Wang and E. H. Adelson, "Representing Moving Images with Layers", IEEE Transactions on Image Processing, Special Issue on Image Sequence Compression, Vol. 3, No. 5, p. 625-638, September 1994.

[17] F. Marqués, V. Vera, and A. Gasull, "A Hierarchical Image Sequence Model for Segmentation: Application to Object-based Sequence Coding", Proc. VCIP 1994, SPIE Visual Communications & Image Processing, p. 554-563, Chicago, IL, USA, October 1994.

[18] C. Gu and M. Kunt, "Very Low Bit-Rate Video Coding using Multi-Criterion Segmentation", Proc. ICIP 1994, IEEE International Conference on Image Processing, Vol. 2, p. 418-422, Texas, USA, November 1994.

[19] J. Benois, L. Wu, and D. Barba, "Joint Contour-based and Motion-based Image Sequences Segmentation for TV Image Coding at Low Bit Rate", Proc: VCIP 1994, SPIE Visual Communications & Image Processing, p. 1074-1085, Chicago, IL, USA, September 1994.

[20] P. Salembier et al., "Segmentation-based Video Coding System allowing the Manipulation of Objects", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 7, No. 1, p. 60-74, February 1997.

[21] S.-Y. Yoon and E. H. Adelson, "Subband Texture Synthesis for Image Coding", Proc. SPIE on Human Vision and Electronic Imaging III, Vol. 3299, p. 489-497, San Jose, CA, USA, January 1998.

[22] A. Dumitraş and B. G. Haskell, "An Encoder-Decoder Texture Replacement Method with Application to Content-based Movie Coding", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 14, No. 6, p. 825-840, June 2004.

[23] E. Z. Flores and J. C. Rolon, "A Method for Bit-Rate Reduction of Compressed Video using Texture Analysis/Synthesis", IEEE Workshop on Signal Processing Systems Design and Implementation, p. 467-472, Athens, Greece, November 2005.

[24] M. Bosch, F. Zhu, and E. Delp, "Spatial Texture Models for Video Compression", Proc. ICIP 2007, IEEE International Conference on Image Processing, Vol. 1, p. 93-96, San Antonio, TX, USA, September 2007.

[25] C. Zhu, X. Sun, F. Wu, and H. Li, "Video Coding with Spatio-Temporal Texture Synthesis", Proc. ICME 2007, IEEE International Conference on Multimedia and Expo, p. 112-115, Beijing, China, July 2007.

[26] D. J. Heeger and J.R. Bergen, „Pyramid–based Texture Analysis/Synthesis", Computer Graphics Proc. (SIGGRAPH 95), p. 229-238, Los Angeles, CA, USA, August 1995.

[27] E. Defez, A. G. Law, A. Rezazadeh, and X. D.Yang, "Key Frame Tweening-Simple Polynomial Computations", Proc. ICARCV 1994, IEEE International Conference on Automation, Robotics and Computer Vision, Vol. 1, p. 177-181, Singapore, November 1994.

[28] A. Smolić, Y. Vatis, H. Schwarz, and T. Wiegand, "Improved H.264/AVC Coding using Long-Term Global Motion Compensation", Proc. VCIP 2004, SPIE Visual Communications & Image Processing, p. 343-354, San Jose, CA, USA, January 2004.

[29] E. Steinbach, T. Wiegand, and B. Girod, "Using Multiple Global Motion Models for Improved Block-based Video Coding", Proc. ICIP 1999, IEEE International Conference on Image Processing, Vol. 2, p. 56-60, Kobe, Japan, October 1999.

[30] R. Forchheimer, O. Fahlander, and T. Kronander, "Low Bit-Rate Coding through Animation", Proc. PCS 1983, p. 113-114, Davis, CA, USA, March 1983.

[31] B. Girod, "Image Sequence Coding using 3D Scene Models", Proc. VCIP 1994, Vol. 2308, p. 1576-1591, Chicago, IL, USA, September 1994.

[32] D. E. Pearson, "Developments in Model-based Video Coding", Proc. of the IEEE, Vol. 83, No. 6, p. 892-906, June 1995.

[33] A. Gupta, R. Jain, "Visual Information Retrieval", Communications of the ACM, Vol.40, No.5, p. 70-79, May 1997.

[34] S. F. Chang, J. R. Smith, M. Beigi, and A. Benitez, "Visual Information Retrieval from Large Distributed Online Repositories", Communications of the ACM, Special Issue on Visual Information Retrieval, Vol. 40, No. 12, p. 12-20, December 1997.

[35] Y. Rui, T. S. Huang, and S. F. Chang, "Image Retrieval: Current Techniques, Promising Directions, and Open Issues", Journal of Visual Communication and Image Representation, Vol. 10, No.1, p. 39-62, March 1999.

[36] M. S. Lew, "Principles of Visual Information Retrieval", ISBN 1-85233-381-2, Springer, London Berlin Heidelberg, 2001.

[37] M. Höynck, T. Auweiler, and J.-R. Ohm, "Application of MPEG-7 Descriptors for Content-based Indexing of Sports Videos", Proc. VCIP 2003, Vol. 5150, p. 1317-1328, Lugano, Switzerland, July 2003.

[38] P. Brodatz, "Textures: A Photographic Album for Artists and Designers", ISBN 0-486-40699-7, Dover Publications, 1996.

[39] J. Portilla and E. P. Simoncelli, "A Parametric Texture Model based on Joint Statistics of Complex Wavelet Coefficients", International Journal of Computer Vision, Vol. 40, No. 1, p. 49-71, December 2000.

[40] R. Wilson and M. Spann, "Image Segmentation and Uncertainty", Pattern Recognition and Image Processing Series, Research Studies Press Ltd, England, 1988.

[41] M. J. Black and P. Anandan, "The Robust Estimation of Multiple Motions: Parametric and Piecewise-smooth Flow Fields", Elsevier Computer Vision and Image Understanding, Vol. 63, No. 1, p. 75-104, January 1996.

[42] K. Fu and J. Mui, "A Survey on Image Segmentation", Pattern Recognition, Vol 13, No. 1, p. 3-16, 1981.

[43] R. Haralick and L. Shapiro, "Image Segmentation Techniques", Computer Vision, Graphics and Image Processing, Vol. 29, p. 100-132, 1985.

[44] N. Pal and S. Pal, "A Review on Image Segmentation Techniques", Pattern Recognition, Vol 26, p. 1277-1294, 1993.

[45] P. K. Sahoo et al., “A Survey of Thresholding Techniques”, Computer Vision, Graphics and Image Processing, Vol. 41, p. 233-260, 1988.

[46] C. H. Chen, „On the Statistical Image Segmentation Techniques“, Proc. CPRI 1981, IEEE Conference on Pattern Recognition and Image Processing, p. 262-266, 1981.

[47] H. D. Cheng, X. H. Jiang, Y. Sun, and Jingli Wang, “Color Image Segmentation: Advances and Prospects”, Pattern Recognition, Vol. 34, p. 2259-2281, 2001.

[48] S. Makrogiannis, G. Economou, and S. Fotopoulos, “A Region Dissimilarity Relation that Combines Feature-Space and Spatial Information for Color Image Segmentation”, IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics, Vol. 35, No. 1, p. 44-53, February 2005.

[49] M. Bicego, V. Murino, and M. A. T. Figueiredo, “Similarity-based Clustering of Sequences using Hidden Markov Models” Machine Learning and Data Mining, p. 86-95, LNAI 2734, Springer, New York, USA, 2003.

[50] J. Freixenet et al., “Yet Another Survey on Image Segmentation: Region and Boundary Information Integration”, Proc. ECCV 2002, European Conference on Computer Vision, p. 408-422, 2002.

[51] T. Pavlidis and Y. Liow, “Integrating Region Growing and Edge Detection”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 12, p. 225-233, 1990.

[52] P. Bonnin, J. Blanc Talon, J. Hayot, and B. Zavidovique, “A New Edge Point/Region Cooperative Segmentation Deduced from a 3D Scene Reconstruction Application”, Proc. SPIE 33rd Annual International Symposium on Optical & Optoelectronic Applied Science & Engineering, Applications of Digital Image Processing XII, Vol. 1153, p. 579-591, San Diego, CA, USA, August 1989.

[53] Y. Xiaohan et al., “Image Segmentation Combining Region Growing and Edge Detection”, Proc. ICPR 1992, International Conference on Pattern Recognition, Vol. C, p. 481-484, The Hague, Netherlands, 1992.

[54] D. Sinclair, “Voronoi Seeded Colour Image Segmentation”, Technical Report 3, AT&T Laboratories Cambridge, 1999.

[55] A. Gagalowicz and O. Monga, “A New Approach for Image Segmentation”, Proc. ICPR 1986, International Conference on Pattern Recognition, p. 265-267, Paris, France, 1986.

[56] M. Spann and R. Wilson, “A Quad-Tree Approach to Image Segmentation which Combines Statistical and Spatial Information”, Pattern Recognition, Vol 18, Nos. 3/4, p. 257-269, 1985.

[57] R. Wilson and M. Spann, “Finite Prolate Spheroidal Sequences and their Applications II: Image Feature Description and Segmentation”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 10, p. 193-203, 1988.

[58] T. Hsu, J. Kuo, and R. Wilson, “A Multiresolution Texture Gradient Method for Unsupervised Segmentation”, Pattern Recognition, Vol 33, p. 1819-1833, 2000.

[59] F. Chan et al., “Object Boundary Location by Region and Contour Deformation”, Proc. IEE Vision Image and Signal Processing, Vol. 143, p. 353-360, 1996.

[60] A. Siebert, "Dynamic Region Growing", International Conference on Vision Interface, Kelowna, Canada, 1997.

[61] J. Chen, T. N. Pappas, A. Mojsilović, and .B. Rogowitz, "Adaptive Perceptual Color-Texture Image Segmentation", IEEE Transactions on Image Processing, Vol. 14, No. 10, p. 1524-1536, October 2005.

[62] R. Falah, P. Bolon, and J. Cocquerez, "A Region-Region and Region-Edge Cooperative Approach of Image Segmentation", Proc. ICIP 1994, IEEE International Conference on Image Processing, Vol. 3, p. 470-474, Austin, Texas, 1994.

[63] Q. Huang and B. Dom, "Quantitative Methods of Evaluating Image Segmentation", Proc. ICIP 1995, IEEE International Conference on Image Processing, Vol. 3, p. 53-56,Washington DC, USA, 1995.

[64] A. Witkin, "Scale-Space Filtering: A New Approach to Multi-Scale Description", Proc. ICASSP 1984, IEEE International Conference on Acoustics Speech and Signal Processing, Vol. 3, p. 39A.1.1-39A.1.4, San Diego, CA, USA, March 1984.

[65] J.J. Koenderink, "The Structure of Images", Biological Cybernetics, Vol. 50, p. 363-370, 1984.

[66] R. Wilson, H. Knutsson, and G. H. Granlund, "The Operational Definition of the Position of Line and Edge", Proc. ICPR 1982, International Conference on Pattern Recognition, Munich, Germany, September 1982.

[67] S. L. Tanimoto and T. Pavlidis, "A Hierarchical Data Structure for Picture Processing", Computer Graphics and Image Processing, Vol. 4, p. 104-119, June 1975.

[68] E. Littmann and H. Ritter, "Adaptive Color Segmentation – A Comparison of Neural and Statistical Methods", IEEE Transactions on Neural Networks, Vol. 8, No. 1, p. 175-185, January 1997.

[69] B. A. Wandell, "Foundations of Vision", ISBN 0-87893-853-2, Sinauer Association, 1995.

[70] G. Wyszecki and W. S. Stiles, "Color Science: Concepts and Methods, Quantitative Data and Formulae", Wiley, New York, 1982.

[71] R. Ohlander, K. Price, and D. R. Reddy, "Picture Segmentation using a Recursive Region Splitting Method", Computer Graphics and Image Processing, Vol. 8, p. 313-333, 1978.

[72] S. Tominaga, "Color Image Segmentation using Three Perceptual Attributes", Proc. CVPR 1986, IEEE Conference on Computer Vision and Pattern Recognition, p. 628-630, Los Alamos, CA, USA, June 1986.

[73] G. D. Guo, S. Yu, and S. D. Ma, "Unsupervised Segmentation of Color Images", Proc. ICIP 1998, IEEE International Conference on Image Processing, p. 299-302, Chicago, IL, USA, 1998.

[74] M. Celenk, "A Color Clustering Technique for Image Segmentation", Proc. CVGIP 1990,Computer Vision, Graphics, and Image Processing, Vol. 52, No. 3, p. 145-170, November 1990.

[75] Y. Ohta, T. Kanade, and T. Sakai, “Color Information for Region Segmentation”, Computer Graphics and Image Processing, Vol. 13, No. 3, p. 222-241, July 1980.

[76] S. Tominaga, “Color Classification of Natural Color Images”, Color Research and Application, Vol. 17, No. 4, p. 230-239, August 1992.

[77] C. M. Bishop, “Neural Networks for Pattern Recognition”, ISBN 0-198-53864-2, Oxford University Press, 1995.

[78] A. Smolić, “Globale Bewegungsbeschreibung und Video Mosaiking unter Verwendung parametrischer 2-D Modelle, Schätzverfahren und Anwendungen”, PhD Thesis, Aachen University of Technology, Germany, 2001.

[79] A. Del Bimbo, P. Nesi, and J. L. C. Sanz, “Optical Flow Computation using Extended Constraints” , IEEE Transactions on Image Processing, Vol. 5, No. 5, p. 720-739, May 1996.

[80] A. Kumar, A. Tannenbaum, and G.J. Balas, “Optical Flow: A Curve Evolution Approach”, IEEE Transactions on Image Processing, Vol. 5, No. 4, p. 598-610, April 1996.

[81] E. P. Simoncelli, “Distributed Representation and Analysis of Visual Motion”, PhD Thesis, Massachussets Institute of Technology, Cambridge, MA, USA, January 1993.

[82] M. Tistarelli, “Multiple Constraints to Compute Optical Flow”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 18, No. 12, p. 1243-1250, December 1996.

[83] D. Wang, “Unsupervised Video Segmentation Based on Watersheds and Temporal Tracking”, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 8 , No. 5, p. 539-546, 1998.

[84] Y. Deng and B. Manjunath, “Unsupervised Segmentation of Color-Texture Regions in Images and Video”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 23, No. 8, p. 800-810, 2001.

[85] A. Del Bimbo, P. Pala, and L. Tanganelli, “Video Retrieval Based on Dynamics of Color Flows”, Proc. ICPR 2000, International Conference on Pattern Recognition, Vol. 1, p. 851-854, Barcelona, Spain, 2000.

[86] J. L. Potter, “Velocity as a Cue to Segmentation”, IEEE Transactions on Systems, Man and Cybernetics, Vol. 5, p. 390-394, 1975.

[87] J. L. Potter, “Scene Segmentation using Motion Information”, Computer Graphics and Image Processing, Vol. 6, No. 6, p. 558-581, 1977.

[88] A. Spoerri and S. Ullman, “The Early Detection of Motion Boundaries”, Proc. ICCV 1987, First IEEE International Conference on Computer Vision, p. 209-218, London, UK, 1987.

[89] E. C. Hildreth, “The Measurement of Visual Motion”, ACM Distinguished Dissertation Series, MIT Press, Cambridge, MA, USA, 1984.

[90] H.-H. Nagel, G. Socher, H. Kollnig, and M. Otte, “Motion Boundary Detection in Image Sequences by Local Stochastic Test”, Proc. ECCV 1994, European Conference on Computer Vision, Vol. 2, p.305-315, Stockholm, Sweden, May 1994.

[91] I. Overington, “Gradient-based Flow Segmentation and Location of the Focus of Expansion”, Third Alvey Vision Conference, p.860-870, Cambridge University, UK, September 1987.

[92] W. B. Thompson, K. M. Mutch, and V. A. Berzins, “Dynamic Occlusion Analysis in Optical Flow Fields”, IEEE Transactions on Pattern Analysis and Machine Intellligence, Vol. 7, No. 4, p.374-383, July 1985.

[93] W. F. Clocksin, “Perception of Surface Slant and Edge Labels from Optical Flow: A Computational Approach”, Perception, Vol. 9, p. 253-269, 1980.

[94] B. G. Shunck, “Image Flow Segmentation and Estimation by Constraint Line Clustering”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 11, No. 10, p. 1010-1027, 1989.

[95] R. Piroddi and T. Vlachos, “A Simple Framework for Spatio-Temporal Video Segmentation and Delayering using Dense Motion Fields”, IEEE Signal Processing Letters, Vol. 13, No. 7, p. 421-424, July 2006.

[96] M. Hötter and R. Thoma, “Image Segmentation Based on Object Oriented Mapping Parameter Estimation”, EURASIP Signal Processing, Vol. 15, No. 3, p. 315-334, 1988.

[97] N. Diehl, “Object-oriented Motion Estimation and Segmentation in Image Sequences”, EURASIP Signal Processing, Vol. 3, No. 1, p. 23-56, 1991.

[98] R. Mech and M. Wollborn, “A Noise Robust Method for 2D Shape Estimation of Moving Objects in Video Sequences Considering a Moving Camera”, EURASIP Signal Processing, Vol. 66, No. 2, p. 203-217, 1998.

[99] R. Thoma and M. Bierling, “Motion Compensating Interpolation Considering Covered and Uncovered Background”, EURASIP Signal Processing, Vol. 1, No. 2, p. 191-212, 1989.

[100] S. N. Jayaramamurthy and R. Jain, “An Approach to the Segmentation of Textured Dynamic Scenes”, Computer Vision, Graphics and Image Processing, Vol. 21, p. 239-261, 1983.

[101] T. Aach, A. Kaup, and R. Mester, “Statistical Model-based Change Detection in Moving Video”, EURASIP Signal Processing, Vol. 31, No. 2, p. 165-180, 1993.

[102] R. Jain and H.-H. Nagel, “On the Analysis of Accumulative Difference Pictures from Image Sequences of Real World Scenes”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 1, No. 2, p. 206-214, April 1979.

[103] R. Jain, “Segmentation of Frame Sequences Obtained by a Moving Observer”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 6, No. 5, p. 624-629, September 1984.

[104] E. Rignot and J. van Zyl, “Change Detection Techniques for ERS-1 SAR Data”, IEEE Transactions on Geoscience and Remote Sensing, Vol. 31, No. 4, p. 896-906, July 1993.

[105] L. Bruzzone and D. F. Prieto, “Automatic Analysis of the Difference Image for Unsupervised Change Detection”, IEEE Transactions on Geoscience and Remote Sensing, Vol. 38, No. 3, p. 1171-1182, May 2000.

[106] R. Mech, "Description of COST 211 Analysis Model (Version 5.1)", COST 211quat Algorithm Subgroup, Thessaloniki, Greece, October 2001.

[107] Y. Z. Hsu, H.-H. Nagel, and G. Reckers, "New Likelihood Test Methods for Change Detection in Image Sequences", Computer Vision, Graphics and Image Processing, Vol. 26, p. 73-106, 1984.

[108] Z. Jain and Y. Chau, "Optimum Multisensor Data Fusion for Image Change Detection", IEEE Transactions on Systems, Man and Cybernetics, Vol. 25, No. 9, p. 1340-1347, September 1995.

[109] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, "Wallflower: Principle and Practice of Background Maintainance", Proc. ICCV 1999, IEEE International Conference on Computer Vision, p. 255-261, Kerkyra, Greece, 1999.

[110] C. Clifton, "Change Detection in Overhead Imagery using Neural Networks", Applied Intelligence, Vol. 18, No. 2, p. 215-234, 2003.

[111] K. Skifstad and R. Jain, "Illumination Independent Change Detection for Real World Image Sequences". Computer Vision, Graphics and Image Processing, Vol. 46, No. 3, p. 387-399, 1989.

[112] E. Durucan and T. Ebrahimi, "Change Detection and Background Extraction by Linear Algebra", Proc. of the IEEE, Vol. 89, No. 10, p. 1368-1381, October 2001.

[113] T. Aach, L. Dümbgen, R. Mester, and D. Toth, "Bayesian Illumination-invariant Motion Detection", Proc. ICIP 2001, IEEE International Conference on Image Processing, p. 640-643, Thessaloniki, Greece, October 2001.

[114] L. Li and M. K. H. Leung, "Integrating Intensity and Texture Differences for Robust Change Detection", IEEE Transactions on Image Processing, Vol. 11, No. 2, p. 105-112, February 2002.

[115] R. J. Radke, S. Andra, O. Al-Kofahi, and B. Roysam, "Image Change Detection Algorithms: A Systematic Survey", IEEE Transactions on Image Processing, Vol. 14, No. 3, p. 294-307, March 2005.

[116] T. Jebara, A. Azarbayejani, and A. Pentland, "3D Structure from 2D Motion", IEEE Signal Processing Magazine, Vol. 16, No. 3, p. 66-84, May 1999.

[117] G. S. Bestor, "Recovering Feature and Observer Position by Projected Error Refinement", PhD Thesis, University of Wisconsin-Madison, WI, USA, 1998.

[118] W. J. MacLean, "Recovery of Egomotion and Segmentation of Independent Object Motion using the EM-Algorithm", PhD Thesis, University of Toronto, Canada, 1996.

[119] G. Adiv, "Determining Three-dimensional Motion and Structure from Optical Flow Generated by Several Moving Objects", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 7, No. 4, p. 384-401, July 1985.

[120] D. Murray and B. Buxton, "Scene Segmentation from Visual Motion using Global Optimization", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 9, No. 2, p. 220-228, March 1987.

[121] M. M. Chang, A. M. Tekalp, and M. I. Sezan, "An Algorithm for Simultaneous Motion Estimation and Segmentation", Proc. ICASSP 1994, IEEE International

Conference on Acoustics Speech and Signal Processing, Vol. 5, p. 221-224, Adelaide, Australia, April 1994.

[122] C. Stiller, "Object-based Estimation of Dense Motion Fields", IEEE Transactions on Image Processing, Vol. 6, No. 2, p. 234-250, February 1997.

[123] C. C. Dorea, M. Pardàs, and F. Marqués, "A Motion-based Binary Partition Tree Approach to Video Object Segmentation", Proc. ICIP 2005, IEEE International Conference on Image Processing, Vol. 2, p. 430-433, Genova, Italy, September 2005.

[124] P. De Smet and I. Bruyland, "On a New Motion Estimation and Segmentation Framework for Digital Video Processing", Proc. DCV 2001, IEEE International Workshop on Digital and Computational Video, p. 69-76, Los Alamitos, CA, USA, February 2001.

[125] M. Gelgon and P. Bouthemy, "A Region-level Motion-based Graph Representation and Labeling for Tracking a Spatial Image Partition", Elsevier Pattern Recognition, Vol. 33, p. 725-740, 2000.

[126] V. Mezaris, I. Kompatsiaris, and M. G. Strintzis, "Video Object Segmentation using Bayes-based Temporal Tracking and Trajectory-based Region Merging", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 14, No. 6, p. 782-795, June 2004.

[127] F. Moscheni, S. Bhattacharjee, and M. Kunt, "Spatiotemporal Segmentation Based on Region Merging", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 20, No. 9, p. 897-915, September 1998.

[128] D. Zhang and G. Lu, "Segmentation of Moving Objects in Image Sequence: A Review", Circuits Systems Signal Processing, Vol. 20, No.2, p. 143-183, 2001.

[129] T. Meier, "Segmentation for Video Object Plane Extraction and Reduction of Coding Artifacts", PhD Thesis, University of West Australia, Australia, 1998.

[130] J. G. Choi, S.-W. Lee, and S.-D. Kim, "Spatio-Temporal Video Segmentation using a Joint Similarity Measure", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 7, No. 2, p. 279-286, April 1997.

[131] A. Alatan et al., "Image Sequence Analysis for Emerging Interactive Multimedia Services – The European COST 211 Framework", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 8, No. 7, p. 802-813, November 1998.

[132] H. Greenspan, J. Goldberger, and A. Mayer, "A Probabilistic Framework for Spatio-Temporal Video Representation and Indexing", Proc. ECCV 2002, European Conference on Computer Vision, Vol. 4, p. 461-475, LNCS 2353, Springer, Berlin, Germany, 2002.

[133] D. De Menthon, "Spatio-Temporal Segmentation of Video by Hierarchical Mean Shift Analysis", Proc. SMVP 2002, Statistical Methods in Video Processing Workshop, Copenhagen, Denmark, 2002.

[134] L. Liu and G. Fan, "Combined Key-Frame Extraction and Object-based Video Segmentation", Transactions on Circuits and Systems for Video Technology, Vol. 15, No. 7, p. 869-884, July 2005.

[135] Y. Wang, K.-F. Loe, T. Tan, and J.-K. Wu, "Spatiotemporal Video Segmentation Based on Graphical Models", IEEE Transactions on Image Processing, Vol. 14, No. 7, p. 937-947, July 2005.

[136] J. Zhu, S. C. Schwartz, and B. Liu, "A Transform Domain Approach to Real-Time Foreground Segmentation in Video Sequences", Proc. ICASSP 2005, IEEE International Conference on Acoustics Speech and Signal Processing, Vol. 2, p. 685-688, Philadelphia, PA, USA, March 2005.

[137] N. O'Connor, T. Adamek, S. Sav, N. Murphy, and S. Marlow, "Qimera: A Software Platform for Video Object Segmentation and Tracking", International Workshop on Image Analysis for Multimedia Interactive Services, WIAMIS 2003, London, U.K., April 2003.

[138] T. Aach, A. Kaup, and R. Mester, "Change Detection in Image Sequences using Gibbs Random Fields: A Bayesian Approach", International Workshop on Intelligent Signal Processing and Communication Systems, Sendai, Japan, p. 56-61, October 1993.

[139] Y. Rubner, C. Tomasi, and L. Guibas, "A Metric for Distributions with Applications to Image Databases", Proc. ICCV'98, IEEE International Conference on Computer Vision, pp.207-214, Bombay, India, 1998.

[140] J. A. Hanley and B. J. Mc Neil, "The Meaning and Use of the Area under the Receiver Operating Characteristic (ROC) Curve", Radiology, Vol. 1, No. 143, p. 29-36, 1982.

[141] Z. Bar-Joseph, R. El-Yaniv, D. Lischinski, and M. Werman, "Texture Mixing and Texture Movie Synthesis using Statistical Learning", IEEE Transactions on Visualization and Computer Graphics, Vol. 7, No. 2, p. 120-135, 2001.

[142] A. Kokaram, "A Statistical Framework for Picture Reconstruction using 2D AR Models", Elsevier Image and Vision Computing, Vol. 22, No. 2, p. 165-171, 2004.

[143] L.-Y. Wei and M. Levoy, "Fast Texture Synthesis using Tree-structured Vector Quantization", Computer Graphics Proc. (SIGGRAPH 00), p. 479-488, New Orleans, LA, USA, July 2000.

[144] A. A. Efros and T. K. Leung, "Texture Synthesis by Non-parametric Sampling", Proc. ICCV 99, IEEE International Conference on Computer Vision, p. 1033-1038, Corfu, Greece, September 1999.

[145] M. Szummer and R. Picard, "Temporal Texture Modeling", Proc. ICIP 96, IEEE International Conference on Image Processing, Vol. 3, p. 823-826, Lausanne, Switzerland, 1996.

[146] S. Soatto, G. Doretto, and Y. Wu, "Dynamic Textures", Proc. ICCV 2001, IEEE International Conference on Computer Vision, Vol. 2, p. 439-446, Vancouver, Canada, July 2001.

[147] Y. Wang and S. Zhu, "A Generative Method for Textured Motion: Analysis and Synthesis", Proc. ECCV 2002, European Conference on Computer Vision, Vol. 1, p. 583-598, Copenhagen, Denmark, 2002.

[148] C.-B. Liu, R.-S. Lin, and N. Ahuja, "Modeling Dynamic Textures using Subspace Mixtures", Proc. ICME 2005, IEEE International Conference on Multimedia and Expo, p. 1378-1381, Amsterdam, Netherlands, July 2005.

[149] J. S. DeBonet, “Multiresolution Sampling Procedure for Analysis and Synthesis of Texture Images”, Computer Graphics Proc. (SIGGRAPH 97), p. 361-368, Los Angeles, CA, USA, August 1997.

[150] M. Ashikhmin, “Synthesizing Natural Textures”, ACM Symposium on Interactive 3D Graphics, p. 217-226, Chapel Hill, NC, USA, 2001.

[151] L. Liang, C. Liu, Y.-Q. Xu, B. Guo, and H.-Y. Schum, “Real-time Texture Synthesis by Patch-based Sampling”, ACM Transactions on Graphics, Vol. 20, No. 3, p. 127-150, July 2001.

[152] A. A. Efros and W. T. Freeman, “Image Quilting for Texture Synthesis and Transfer”, Computer Graphics Proc. (SIGGRAPH 01), p. 341-346, Los Angeles, CA, USA, August 2001.

[153] A. Schödl, R. Szeliski, D. H. Salesin, and I. Essa, “Video Textures“, Computer Graphics Proc. (SIGGRAPH 00), p. 489-498, New Orleans, LA, USA, July 2000.

[154] V. Kwatra, A. Schödl, I. Essa, G. Turk, and A. Bobick, “Graphcut Textures: Image and Video Synthesis using Graph Cuts”, Computer Graphics Proc. (SIGGRAPH 03), p. 277-286, San Diego, CA, USA, July 2003.

[155] A. Criminisi, P. Pérez, and K. Toyama, “Region Filling and Object Removal by Exemplar-based Image Inpainting”, IEEE Transactions on Image Processing, Vol. 13, No. 9, p. 1200-1212, September 2004.

[156] A. Nealen and M. Alexa, “Hybrid Texture Synthesis”, Eurographics Workshop on Rendering, p. 97-105, Leuven, Belgium, June 2003.

[157] A. Smolić and J.-R. Ohm, “Robust Global Motion Estimation using a Simplified M-Estimator Approach”, Proc. ICIP 2000, IEEE International Conference on Image Processing, Vancouver, Canada, September 2000.

[158] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, “Network Flows: Theory, Algorithms, and Applications”, ISBN 0-13-617549-x, Prentice Hall, 1993.

[159] S. N. Sinha, “Graph Cut Algorithms in Vision, Graphics and Machine Learning – An Integrative Paper”, UNC Chapel Hill, November 2004.

[160] Y. Boykov, O. Veksler, and R. Zabih, “Fast Approximate Energy Minimization via Graph Cuts”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 23, No. 11, p. 1222-1239, 2001.

[161] V. Kolmogorov, “Graph-based Algorithms for Scene Reconstruction from Two or More Views”, PhD Thesis, Cornell University, USA, 2004.

[162] L. Ford and D. Fulkerson, “Flows in Networks”, Princeton University Press, 1962.

[163] A. V. Goldberg and R. E. Tarjan, “A New Approach to the Maximum-Flow Problem”, Journal of the Association for Computing Machinery, Vol. 35, No. 4, p. 921-940, October 1988.

[164] X. Qin and Y.-H. Yang, “Theoretical Analysis of Graphcut Textures”, Technical Report 05-26, Department of Computer Science, University of Alberta, 2005.

[165] P. J. Burt and E. H. Adelson, “A Multiresolution Spline with Application to Image Mosaics”, ACM Transactions on Graphics, Vol. 2, No. 4, p. 217-236, October 1983.

[166] A. N. Hirani and T. Totsuka, "Combining Frequency and Spatial Domain Information for Fast Interactive Image Noise Removal", Computer Graphics Proc. (SIGGRAPH 96), p. 269-276, New Orleans, LA, USA, August 1996.

[167] M. Bertalmio, G. Shapiro, V. Caselles, and C. Ballester, "Image Inpainting", Computer Graphics Proc. (SIGGRAPH 00), p. 417-424, New Orleans, LA, USA, July 2000.

[168] T. Georgiev, "Image Reconstruction Invariant to Religthing", Proc. Eurographics 2005, European Association for Computer Graphics, Dublin, Ireland, August-September 2005.

[169] S. Rane, G. Sapiro, and M. Bertalmio, "Structure and Texture Filling-In of Missing Image Blocks in Wireless Transmission and Compression Applications", IEEE Transactions on Image Processing, Vol. 12, No. 3, p. 296-303, March 2003.

[170] K. A. Patwardhan, G. Sapiro, and M. Bertalmio, "Video Inpainting Under Constrained Camera Motion", IEEE Transactions on Image Processing, Vol. 16, No. 2, p. 545-553, February 2007.

[171] S. Winkler, "Issues in Vision Modeling for Perceptual Video Quality Assessment", Elsevier Signal Processing, Vol. 78, p. 231-252, 1999.

[172] W. Y. Zou, "Performance Evaluation: From NTSC to Digitally Compressed Video", SMPTE J., Vol. 103, No. 12, p. 795-800, December 1994.

[173] EIA-250-B, "Electrical Performance Standard for Television Relay Facilities", Electronic Industries Association, Washington, D.C., USA, 1976.

[174] M. H. Pinson and S. Wolf, "A New Standardized Method for Objectively Measuring Video Quality", IEEE Transactions on Broadcasting, Vol. 50, No. 3, p. 312-322, September 2004.

[175] American National Standards Institute T1.801.02, "Digital Transport of Video Teleconferencing/Video Telephony Signals – Performance, Terms, Definitions, and Examples", 1995.

[176] M. Yuen and H. R. Wu, "A Survey of Hybrid MC/DPCM/DCT Video Coding Distortions", EURASIP Signal Processing, Vol. 70, No. 3, p. 247-278, 1998.

[177] S. D. Voran and S. Wolf, "The Development and Evaluation of an Objective Video Quality Assessment System that Emulates Human Viewing Panels", International Broadcasting Convention, Amsterdam, The Netherlands, 1992.

[178] ITU-R WG6Q 6/39-E, "Objective Perceptual Video Quality Measurement Techniques for Standard Definition Digital Broadcast Television in the Presence of a Full Reference", October 2003.

[179] Z. Wang, A. Conrad Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image Quality Assessment: From Error Visibility to Structural Similarity", IEEE Transactions on Image Processing, Vol. 13, No. 4, p. 600-612, April 2004.

[180] S. Olsson, M. Stroppiana, and J. Baïna, "Objective Methods for Assessment of Video Quality: State of the Art", IEEE Transactions on Broadcasting, Vol. 43, No. 4, p. 487-495, December 1997.

[181] J. Guo, M. Van Dyke-Lewis, and H. R. Myler, "Gabor Difference Analysis of Digital Video Quality", IEEE Transactions on Broadcasting, Vol. 50, No. 3, p. 302-311, September 2004.

[182] M. Carnec, P. Le Callet, and D. Barba, "Visual Features for Image Quality Assessment with Reduced Reference", Proc. ICIP 2005, IEEE International Conference on Image Processing, Vol. 1, p. 421-424, Genova, Italy, September 2005.

[183] A. B. Watson and J. Malo, "Video Quality Measures Based on the Standard Spatial Observer", Proc. ICIP 2002, IEEE International Conference on Image Processing, Vol. 3, p. 41-44, Rochester, New York, USA, 2002.

[184] E. P. Ong, W. Lin, Z. Lu, and S. Yao, "Colour Perceptual Video Quality Metric", Proc. ICIP 2005, IEEE International Conference on Image Processing, Vol. 3, p. 1172-1175, Genova, Italy, September 2005.

[185] E. P. Ong, X. Yang, W. Lin, Z. Lu, and S. Yao, "Video Quality Metric for Low Bitrate Compressed Video", Proc. ICIP 2004, IEEE International Conference on Image Processing, p. 3531-3534, Singapore, October 2004.

[186] Z. Wang and E. P. Simoncelli, "An Adaptive Linear System Framework for Image Distortion Analysis", Proc. ICIP 2005, IEEE International Conference on Image Processing, Vol. 3, p. 1160-1163, Genova, Italy, September 2005.

[187] P. N. Gardiner, M. Ghanban, D. E. Pearson, and K. T. Tan, "Development of a Perceptual Distortion Meter for Digital Video", IEE International Broadcasting Convention, p. 493-497, Amsterdam, The Netherlands, 1997.

[188] O. Kwon and C. Lee, "Objective Method for Assessment of Video Quality Using Wavelets", Proc. ISIE 2001, IEEE International Symposium on Industrial Electronics, p. 292-295, Pusan, Korea, 2001.

[189] L. Lu, Z. Wang, A. Bovik, and J. Khouloheris, "Full-Reference Video Quality Assessment Considering Structural Distortion and No-Reference Quality Evaluation of MPEG Video", Proc. ICME 2002, IEEE International Conference on Multimedia and Expo, Vol. 1, p. 61-64, Lausanne, Switzerland, August 2002.

[190] Y. Kato, A. Honda, and K. Hokozaki, "An Analysis of Relationship Between Video Contents and Subjective Video Quality for Intenet Broadcasting", Proc. AINA 2005, IEEE International Conference on Advanced Information Networking, and Applications, Vol. 2, p. 31-34, Taipei, Taiwan, 2005.

[191] Y. Inazumi, Y. Horita, K. Kotani, and T. Murai, "Quality Evaluation Method Considering Time Transition of Coded Video Quality", Proc. ICIP 1999, IEEE International Conference on Image Processing, Vol. 4, p. 338-342, Kobe, Japan, 1999.

[192] Z. Yu, H. R. Wu, S. Winkler, and T. Chen, "Vision-Model-Based Impairment Metric to Evaluate Blocking Artifacts in Digital Video", Proc. of IEEE, Vol. 90, No. 1, p. 154-169, January 2002.

[193] H. R. Wu and M. Yuen, "A Generalized Block-Edge Impairment Metric for Video Coding", IEEE Signal Processing Letters, Vol. 4, No. 11, p. 317-320, November 1997.

[194] V. Ojansivu, O. Silvén, and R. Huotari, "A Technique for Digital Video Quality Evaluation", Proc. ICIP 2003, IEEE International Conference on Image Processing, Barcelona, Spain, 2003.

[195] S. Bin and C. Yilin, "Control Strategy of Subjective Video Quality Over Error-Prone Channels", Proc. ICCT 2000, International Conference on Communication Technology, Vol. 1, p. 962-965, Beijing, China, 2000.

[196] F. Yang, S. Wan, Y. Chang, and H. R. Wu, "A Novel Objective No-Reference Metric for Digital Video Quality Assessment", IEEE Signal Processing Letters, Vol. 12, No. 10, p. 685-688, October 2005.

[197] N. Montard and P. Brétillon, "Objective Quality Monitoring Issues in Digital Broadcasting Networks", IEEE Transactions on Broadcasting, Vol. 51, No. 3, p. 269-275, September 2005.

[198] B. Girod, "What's Wrong with Mean-Squared Error", Digital Images and Human Vision, Ed. Cambridge, MA: MIT Press, p. 207-220, 1993.

[199] P. C. Teo and D. J. Heeger, "Perceptual Image Distortion", Proc. SPIE, Vol. 2179, p. 127-141, 1994.

[200] S. Pefferkorn and J.-L. Blin, "Perceptual Quality Metric of Color Quantization Errors on Still Images", Proc. SPIE, Vol. 3299, p. 210-220, San Jose, CA, USA, 1998.

[201] Z. Wang, A. C. Bovik, and L. Lu, "Why is Image Quality Assessment so Difficult", Proc. ICASSP 2002, IEEE International Conference on Acoustics Speech and Signal Processing, Vol. 4, p. 3313-3316, Orlando, FL, USA, May 2002.

[202] N. S. Jayant and P. Noll, "Digital Coding of Waveforms: Principles and Applications to Speech and Video", ISBN 0-132-11913-7, Prentice Hall Signal Processing Series, 1984.

[203] A. B. Watson, "The Cortex Transform: Rapid Computation of Simulated Neural Images", Computer Vision, Graphics and Image Processing, Vol. 39, No. 3., p. 311-327, 1987.

[204] S. Daly, "The Visible Difference Predictor: An Algorithm for the Assessment of Image Fidelity", Digital Images and Human Vision, Ed. Cambridge, MA: MIT Press, p. 179-206, 1993.

[205] J. Lubin, "The Use of Psychophysical Data and Models in the Analysis of Display System Performance", Digital Images and Human Vision, Ed. Cambridge, MA: MIT Press, p. 163-178, 1993.

[206] D. A. Silverstein and J. E. Farrell, "The Relationship Between Image Fidelity and Image Quality", Proc. ICIP 1996, IEEE International Conference on Image Processing, p. 881-884, Lausanne, Switzerland, September 1996.

[207] D. R. Fuhrmann, J. A. Baro, and J. R. Cox Jr., "Experimental Evaluation of Psychophysical Distortion Metrics for JPEG-Encoded Images," SPIE Journal of Electronic Imaging, Vol. 4, No. 4, p. 397-406, October 1995.

[208] ITU-R BT-500-11, "Methodology for Subjective Assessment of the Quality of Television Pictures", ITU-R Recommendations.

[209] H. Hoffmann, “HDTV - EBU Format Comparisons at IBC 2006”, EBU Technical Review, p. 1-8, October 2006.

[210] Standards Committee T1A1, “Proposal for an ANSI Standard Specification of Video Performance Terms and Definitions”, T1A1.5/95-108, January 1995.

[211] D. Marr, “Vision: A Computational Investigation into the Human Representation and Processing of Visual Information”, ISBN 0-716-71567-8, Henry Holt and Co., 1982.

[212] X. Ran and N. Farvardin, “A Perceptually Motivated Three-Component Image Model – Part I: Description of the Model”, IEEE Transactions on Image Processing, Vol. 4, No. 4, p. 401-415, April 1995.

[213] ITU-R WG6Q 6/39-E, "Final Report from the Video Quality Experts Group on the Validation of Objective Models of Video Quality Assessment, Phase II", August 2003.

[214] ISO/IEC JTC 1/SC 29/WG 11 N3671, “Call for Proposals for New Tools to further Improve Video Coding Efficiency”, La Baule, France, October 2000.

[215] W. L. Hays, “Statistics for Psychologists”, New York: Holt, Rinehard, and Winston, 1963.

[216] C. E. Shannon, “A Mathematical Theory of Communication”, Bell System Technical Journal, Vol. 27, p. 379-423, July 1948.

[217] C. E. Shannon, “Coding Theorems for a Discrete Source with a Fidelity Criterion”, IRE National Convention Record, Part 4, p. 142-163, 1959.

[218] A. Ortega and K. Ramchandran, “Rate-Distortion Methods for Image and Video Compression”, IEEE Signal Processing Magazine, Vol. 15, No. 6, p. 23-50, November 1998.

[219] G. Schuster and A. Katsaggelos, “Rate-Distortion based Video Compression”, Kluwer, Boston, 1997.

[220] H. Everett, “Generalized Lagrange Multiplier Method for Solving Problems of Optimum Allocation Resources”, Operations Research, Vol. 11, p. 399-417, 1963.

[221] G. J. Sullivan and T. Wiegand, “Rate-Distortion Optimization for Video Compression”, IEEE Signal Processing Magazine, Vol. 15, No. 6, p. 74-90, November 1998.

[222] H. Wang, G. M. Schuster, and A. K. Katsaggelos, “Rate-Distortion Optimal Bit Allocation for Object-based Video Coding”, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 15, No. 9, p. 1113-1123, September 2005.

[223] ITU-T Rec. H.261, “Video Codec for Audiovisual Services at px64 kbits/s”, v1: November 1990, v2: March 1993.

[224] ITU-T Rec. H.262 & ISO/IEC 13818-2 MPEG-2, “Generic Coding of Moving Pictures and Associated Audio Information – Part 2: Video”, November 1994 (with several subsequent amendments and corrigenda).

[225] ITU-T Rec. H.263, “Video Coding for Low Bit Rate Communication”; v1: November 1995, v2: January 1998, v3: November 2000.

[226] ISO/IEC 11172 MPEG-1, "Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1.5 Mbit/s – Part 2: Video", November 1993.

[227] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC Video Coding Standard", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 13, No. 7, p. 560-576, July 2003.

[228] G. J. Sullivan, P. Topiwala, and A. Luthra, "The H.264/AVC Advanced Video Coding Standard: Overview and Introduction to the Fidelity Range Extensions", Proc. SPIE on Applications of Digital Image Processing, Vol. 5558, p. 454-474, November 2004.

[229] H. Schwarz, D. Marpe, and T. Wiegand, "Analysis of Hierarchical B Pictures and MCTF", Proc. ICME 2006, IEEE International Conference on Multimedia and Expo, Vol. 5, p. 1929-1932, Toronto, Canada, July 2006.