Restoration of Archived Film and Video

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof.ir. K.F. Wakker,
in het openbaar te verdedigen ten overstaan van een commissie,
door het College voor Promoties aangewezen,
op vrijdag 1 oktober te 13.30 uur

door

Peter Michael Bruce VAN ROOSMALEN

elektrotechnisch ingenieur,
geboren te Maastricht.

Dit proefschrift is goedgekeurd door de promotoren:

Prof.dr.ir. J. Biemond
Prof.dr.ir. R.L. Lagendijk

Samenstelling promotiecommissie:

Rector Magnificus	voorzitter
Prof.dr.ir. J. Biemond	Technische Universiteit Delft, promotor
Prof.dr.ir. R.L. Lagendijk	Technische Universiteit Delft, promotor
Prof.dr.ir. P. Noll	Technische Universität Berlin, Duitsland
Prof.dr. A.C. Kokaram	Trinity College, Dublin, Ierland
Prof.dr. I.T. Young	Technische Universiteit Delft
Prof.dr.ir. F.W. Jansen	Technische Universiteit Delft
Prof.dr.ir. A.H.M. van Roermund	Technische Universiteit Delft

Restoration of Archived Film and Video
Van Roosmalen, Peter Michael Bruce
Thesis Delft University of Technology - with ref. - with Summary in Dutch
Printed by Universal Press, Veenendaal
Cover photo: Philip Broos
ISBN 90-901-2792-5

Copyright © 1999 by P.M.B. van Roosmalen

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

Table of Contents

Summary **vii**

List of abbreviations..ix

List of symbols xi

1 Introduction **1**

1.1 Background 1
1.2 Scope 2
1.3 Thesis outline 5

2 Modeling and coding **7**

2.1 Modeling for image restoration...7
  2.1.1 Model selection and parameter estimation 7
  2.1.2 Impairments in old film and video sequences 9
  2.1.3 Influence of artifacts on motion estimation.......................................13
2.2 Image restoration and storage 17
  2.2.1 Brief description of MPEG2..17
  2.2.2 Influence of artifacts on coding efficiency 19

3 Intensity flicker correction **23**

3.1 Introduction...23
3.2 Estimating and correcting intensity flicker in stationary sequences 24
  3.2.1 A model for intensity flicker 24
  3.2.2 Estimating intensity-flicker parameters in stationary scenes 26
  3.2.3 Measure of reliability for the estimated model parameters 28
3.3 Incorporating motion...29
  3.3.1 Estimating global motion with phase correlation 30

3.3.2 Detecting the remaining local motion 30
3.3.3 Interpolating missing parameters 32
3.4 Practical issues 35
3.5 Experiments and results 36
3.5.1 Experiments on artificial intensity flicker 37
3.5.2 Experiments on naturally degraded film sequences 38
3.6 Discussion 44

4 Blotch detection and correction 45

4.1 System for blotch detection and correction 45
4.2 Overview of existing techniques 48
4.2.1 Blotch detection techniques 48
4.2.2 Techniques for motion vector repair 53
4.2.3 Blotch correction techniques 56
4.2.4 Discussion 58
4.3 Improved blotch detection by postprocessing 59
4.3.1 Simplified ROD detector 59
4.3.2 Extracting candidate blotches 60
4.3.3 Removing false alarms due to noise 60
4.3.4 Completing partially detected blotches 63
4.3.5 Constrained dilation for missing details 64
4.3.6 Experimental evaluation 65
4.4 Blotch detection with increased temporal aperture 67
4.5 Fast, good quality interpolation of missing data 69
4.5.1 Interpolating missing data with controlled pasting 71
4.5.2 Practical implementation of controlled pasting 74
4.5.3 Experiments with controlled pasting 76
4.6 Results and discussion 80

5 Noise reduction by coring 81

5.1 Introduction 81
5.2 Noise reduction techniques 82
5.2.1 Optimal linear filtering in the MMSE sense 82
5.2.2 Optimal noise reduction by nonlinear filtering: coring 83
5.2.3 Heuristic coring functions 84
5.3 Coring image sequences in wavelet and subband domains 86
5.3.1 Nondecimated wavelet transform 86
5.3.2 Simoncelli pyramid 89
5.3.3 An extension to three dimensions using wavelets 92
5.3.4 Noise reduction by coring 93
5.3.5 Perfect reconstruction? 95
5.3.6 Experiments and results 96
5.4 MPEG2 for noise reduction 98
5.4.1 Coring I, P, and B frames 99
5.4.2 Determining the DCT coring functions 100
5.4.3 Experiments and results 102
5.5 Discussion 105

6 Evaluation of restored image sequences 107

6.1 Introduction ... 107
6.2 Assessment of restored image sequences ... 108
  6.2.1 Influence of image restoration on the perceived image quality ... 108
  6.2.2 Influence of image restoration on the coding efficiency ... 109
6.3 Experiments and results ... 112
  6.3.1 Test sequences ... 112
  6.3.2 Experiments on image restoration and perceived quality ... 114
  6.3.3 Experiments on image restoration and coding efficiency ... 114
  6.3.4 Discussion of experimental results ... 117
6.4 Discussion ... 117

Appendix A Hierarchical motion estimation 119

Appendix B Derivation of conditionals 121

B.1 Introduction ... 121
B.2 Conditional for AR coefficients ... 122
B.3 Conditional for the prediction error variance ... 124
B.4 Conditional for the direction of interpolation ... 124

Appendix C Optimal quantizers for encoding noisy image sequences 127

Bibliography 131

Samenvatting 139

Acknowledgements 141

Curriculum Vitae 143

Summary

Unique records of historic, artistic, and cultural developments of every aspect of the 20th century are stored in huge stocks of archived moving pictures. Many of these historically significant items are in a fragile state and are in desperate need of conservation and restoration. Preservation of visual evidence of important moments in history and of our cultural past is not only of purely scientific value. Digital broadcast will make many channels available to the home viewer in the near future. These channels require programming, and the huge collection of movies, soaps, documentaries, and quiz shows currently held in store provides a cheap alternative to the high costs of creating new programs. However, re-using old film and video material is only feasible if the visual and audio quality meets the standards expected by the modern viewer.

There is a need for an automated tool for image restoration due to the vast amounts of archived film and video and due to economical constraints. The term *automated* should be stressed because manual image restoration is a tedious and time-consuming process. A project named AURORA was initiated in 1995, stimulated by the European Union ACTS program. The acronym AURORA stands for AUtomated Restoration of ORiginal film and video Archives. The objective of this 3-year project was to create state-of-the-art algorithms in *real-time* hardware for the restoration of old video and film sequences. Such restoration system had to allow for bulk processing, and had to reduce the high costs of manual labor by requiring a minimum of human intervention. At that time, existing commercial restoration tools required much user intervention, and they did not allow for automatic restoration of most common artifacts.

The Delft University of Technology was a partner in the AURORA consortium. This thesis describes the research carried out in Delft in the context of the AURORA project. At Delft, algorithms were developed for correcting three types of artifact common to old film and video sequences, namely intensity flicker, blotches and noise. Intensity flicker is a common artifact in old black-and-white film sequences. It is perceived as unnatural temporal fluctuations in image intensity that do not originate from the original scene. This thesis describes an original, effective method for correcting intensity flicker on the basis of equalizing local intensity mean and variance in a temporal sense.

Blotches are artifacts typically related to film that are caused by the loss of gelatin and dirt particles covering the film. Existing techniques for blotch detection generate many false alarms when high correct detection rates are required. As a result, unnecessary errors that are visually more disturbing than the blotches themselves can be introduced into an image sequence by the interpolators that correct the blotches. This thesis describes techniques to improve the quality of blotch detection results by taking into account the influence of noise on the detection process and by exploiting the spatial coherency within blotches. Additionally, a new, fast, model-based method for good quality interpolation of blotched data is developed. This method is faster than existing model-based interpolators. It is also more robust to corruption in the reference data that is used by the interpolation process.

Coring is a well-known technique for removing noise from still images. The mechanism of coring consists of transforming a signal into a frequency domain and reducing the transform coefficients by the coring function. The inverse transform of the cored coefficients gives the noise-reduced image. This thesis develops a framework for coring image sequences. The framework is based on 3D (2D space and time) image decompositions, which allows temporal information to be exploited. This is preferable to processing each frame independently of the other frames in the image sequence. Furthermore, a method of coring can be imbedded into an MPEG2 encoder with relatively little additional complexity. The MPEG2 encoder then becomes a device for simultaneous noise reduction and image sequence compression. The adjusted encoder significantly increases the quality of the coded noisy image sequences.

Not only does image restoration improve the perceived quality of the film and video sequences, it also, generally speaking, leads to more efficient compression. This means that image restoration gives better quality at fixed bitrates, or, conversely, identical quality at lower bitrates. The latter is especially important in digital broadcasting and storage environments for which the price of broadcasting/storage is directly related to the number of bits being broadcast/stored. This thesis investigates the influence of artifacts on the coding efficiency and it evaluates how much is gained by restoring impaired film and video sequences. It shows that considerable savings in bandwidth are feasible without loss of quality.

List of abbreviations

2AFC	2 Alternatives Forced Choice
2D	Two Dimensional
3D	Three Dimensional
AR	AutoRegressive
CCD	Charge Coupled Device
CP	Controlled Pasting
DCT	Discrete Cosine Transform
DSCQS	Double Stimulus Continuous Quality Scale
DWT	Discrete Wavelet Transform
FIR	Finite Impulse Response
GOP	Group Of Pictures
i.i.d.	Independently and Identically Distributed
ITU	International Telecommunication Union
LMMSE	Linear Minimum Mean Squared Error
MAP	Maximum A Posteriori
MMF	Multistage Median Filter
MPEG	Motion Pictures Expert Group
MRF	Markov Random Field
MMSE	Minimum Mean Squared Error
MSE	Mean Squared Error
RMSE	Root Mean Squared Error
OS	Order Statistics
PAL	Phase Alternating Lines
pdf	Probability Density Function
pmf	Probability Mass Function
PR	Perfect Reconstruction
PSD	Power Spectral Density
PSNR	Peak Signal to Noise Ratio
ROC	Receiver Operator Characteristic
ROD	Rank-Ordered Differences
SA	Simulated Annealing
SAD	Summed Absolute Differences
SDIa	Spike Detection Index-a
SOR	Successive OverRelaxation

SROD	Simplified ROD
SRODex	Extended SROD
SSD	Summed Squared Differences
TM5	Test Model 5
WMSE	Weighted Mean Squared Error
VCR	Video Cassette Recorder
VLC	Variable Length Coding

List of symbols

$\alpha(\boldsymbol{i})$	Multiplicative intensity-flicker model parameter.
$\hat{\alpha}_{m,n}(t)$	Estimate of local multiplicative intensity-flicker model parameter.
$\beta(\boldsymbol{i})$	Additive intensity-flicker model parameter.
$\hat{\beta}_{m,n}(t)$	Estimate of local additive intensity-flicker model parameter.
β	Parameter that controls strength of self organization in MRFs.
γ	Cooling schedule parameter in SA.
ΔQ	Increase in coding efficiency.
$\varepsilon(\boldsymbol{i})$	Difference between true value and estimated value.
κ	Factor that controls bias.
λ	Factor that controls smoothness of SOR results.
μ	Mean value.
σ	Standard deviation.
$\eta(\boldsymbol{i})$	Frame consisting of noise samples, or, depending on the context, a specific pixel in that frame.
ω	Frequency component in a (scale-space) transform domain.
ϖ	Overrelaxation parameter used by SOR.
$\Omega_{m,n}$	Image region.
a, b, c	Parameters that define the generalized gaussian distribution.
$a(\boldsymbol{i})$	Multiplicative intensity-flicker correction parameter.
$b(\boldsymbol{i})$	Additive intensity-flicker correction parameter.
a_k	AR-model coefficient.
$\boldsymbol{a}$	AR-model coefficients placed in a vector.
$\boldsymbol{A}$	AR-model coefficients placed in a matrix.
$c(\boldsymbol{i})$	Frame indicating the intensities of blotched pixels.
$c_{k,l,t}$	Coefficients of a time-varying 2D polynomial.
$d(\boldsymbol{i})$	Binary blotch detection mask.
$d_{k,l}$	Quantizer decision levels.
$e(i)$	Prediction error.
$\boldsymbol{e}$	Prediction error vector.
$\boldsymbol{i}$	Tuple (i, j, t) that represents discrete spatio-temporal coordinates.
h_k	Filter coefficients.
k	Index/number.
l	Index/number.
m	Index/number.
n	Index/number.
N	Number of pixels in a frame.
$N(\omega)$	$\eta(\boldsymbol{i})$ in a transform domain.

$o(\boldsymbol{i})$	Binary mask indicating direction of interpolation for each pixel in the CP method.
O	Set of eight-connected neighbors of $o(\boldsymbol{i})$.
p_k	Reference pixel with index k.
$\boldsymbol{q}_k$	Relative position of a reference pixel with respect to the current pixel.
$q_i(t)$	Global horizontal displacement of a frame t.
$q_j(t)$	Global vertical displacement of a frame t.
r_k	Sample with index k in a set of data ordered by rank.
$r_{k,l}$	Quantizer representation level.
r_{mean}	Rank order mean.
$\boldsymbol{r}_{xx}$	Autocorrelation vector.
R	Risk.
$\boldsymbol{R}_{xx}$	Autocorrelation matrix.
S	Region of support.
S_{xx}	Power Spectral Density.
T	Threshold.
$\boldsymbol{T}$	Temperature parameters used by SA.
$\boldsymbol{v}(\boldsymbol{i})$	Motion vector field.
$W_{m,n}(t)$	Weight for reliability of estimated flicker parameters.
$y(\boldsymbol{i})$	Original, unimpaired frame, or, depending on the context, a specific pixel in that frame.
$\hat{y}(\boldsymbol{i})$	Estimate of original, unimpaired frame, or, depending on the context, a specific pixel in that frame.
$\hat{y}_c(\boldsymbol{i})$	Estimate of original, unimpaired frame after MPEG2 encoding.
$\hat{y}_o(\boldsymbol{i})$	Estimate of original, unimpaired frame prior to MPEG2 encoding.
$y_{mc}(\boldsymbol{i}, t+k)$	Motion compensated frame representing a frame $y(\boldsymbol{i})$ recorded at time t computed from a reference frame recorded at time $t+k$.
$Y(\omega)$	$y(\boldsymbol{i})$ in a (scale-space) frequency transform domain.
$\hat{Y}(\omega)$	$\hat{y}(\boldsymbol{i})$ in a (scale-space) frequency transform domain.
Y_k	DCT coefficient with coefficient number k.
$z(\boldsymbol{i})$	Observed frame, or, depending on the context, a specific pixel in that frame.
$\boldsymbol{z}_+$	Vector with motion compensated observed data from previous, current and next frames.
$z_{mc}(\boldsymbol{i}, t+k)$	Motion compensated frame representing a frame $z(\boldsymbol{i})$ recorded at time t computed from a reference frame recorded at time $t+k$.
$z_c(\boldsymbol{i})$	Observed frame after MPEG2 encoding.
$z_o(\boldsymbol{i})$	Observed frame prior to MPEG2 encoding.
$Z(\omega)$	$z(\boldsymbol{i})$ in a (scale-space) frequency transform domain.

Chapter 1

Introduction

1.1 Background

Unique records of historic, artistic, and cultural developments of every aspect of the 20th century are stored in huge stocks of archived moving pictures. Many of these historically significant items are in a fragile state and are in desperate need of conservation and restoration. Preservation of visual evidence of important moments in history and of our cultural past is not only of purely scientific value. Digital broadcast will make many channels available to the home viewer in the near future. These channels require programming, and the huge collection of movies, soaps, documentaries, and quiz shows currently held in store provides a cheap alternative to the high costs of creating new programs. However, re-using old film and video material is only feasible if the visual and audio quality meets the standards expected by the modern viewer.

If one considers that archived film and video sequences will be preserved by transferring them onto new digital media, there are a number of reasons why these sequences should be restored before renewed storage. First, restoration improves the subjective quality of the film and video sequences (and it thereby increases the commercial value of the film and video documents). Second, restoration generally leads to more efficient compression, i.e., to better quality at identical bitrates, or, conversely, to identical quality at lower bitrates. The latter is especially important in digital broadcasting and storage environments for which the price of broadcasting/storage is directly related to the number of bits being broadcast/stored.

There is a need for an automated tool for image restoration due to the vast amounts of archived film and video and due to economical constraints. The term *automated* should be stressed because manual image restoration is a tedious and time-consuming process. A project named AURORA was initiated in 1995, stimulated by the European Union ACTS program. The acronym AURORA stands for AUtomated Restoration of ORiginal video and film Archives. The objective of this 3-year project was to create state-of-the-art algorithms in *real-time* hardware for the restoration of old video and film sequences. Such restoration system had to allow for bulk processing, and had to reduce the high costs of manual labor by requiring a minimum of human intervention. At that time, existing commercial restoration tools required much user intervention, and they did not allow for automatic restoration of most common artifacts.

1.2 Scope

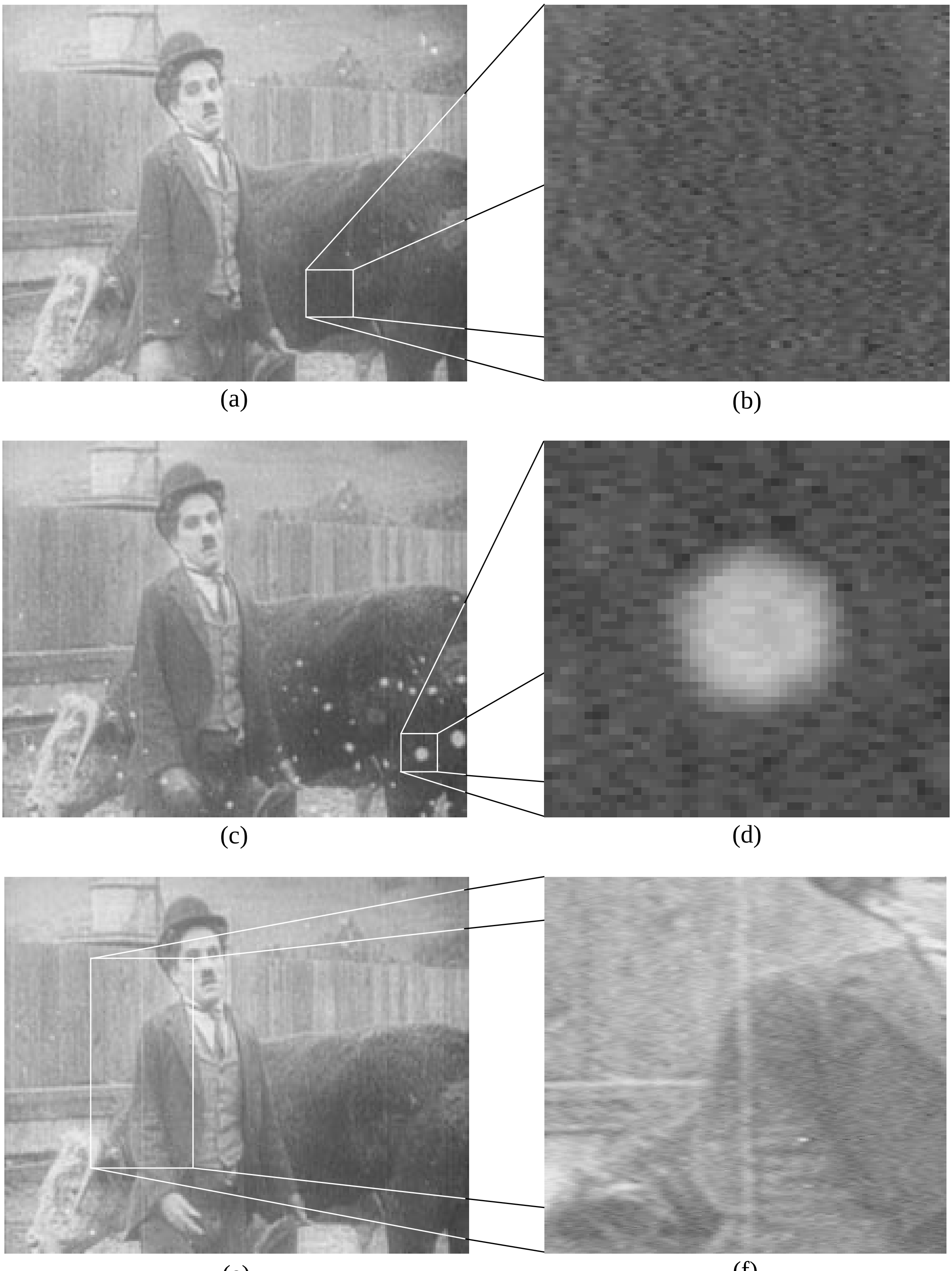
Detecting and restoring selected artifacts from archived film and video material with real-time hardware places constraints on how that material is processed and on the complexity of the algorithms used. It is stressed here that these constraints do not restrict the complexity of the methods for image restoration presented in this thesis, with the exception of the work presented in Chapter 3. Even though much of the work described in this thesis is too complex (meaning too expensive) to be implemented in hardware directly, it gives good insight into the nature of the investigated artifacts. The work presented here gives an upper bound on the quality that can be achieved under relaxed constraints.

The work in this thesis is restricted to black-and-white image sequences for two reasons. First, a large proportion of the films that require restoration is in black and white. Second, most of the algorithms can easily be extended to color, though perhaps in a suboptimal manner. An example of this would be a situation in which a color image sequence is restored by applying the restoration algorithms to the R, G, and B channels separately. Multi channel approaches [6,7] could be taken from the start, at the cost of increased complexity and at the risk of achieving little significant gain compared to what single channel processing already brings.

An inventory of impairments found in old film and video sequences was compiled by AURORA. A list resulted with over 150 entries that indicate the nature of the defects and the frequency of their occurrence. From this list, a number of impairments to be addressed by the AURORA project were selected. The most important are *noise* [1,5,13,22,23,35,46, 69,70,85], *blotches* [26,29,45,47,48,49,63,64,80,82,96], *line scratches* [47,62], *film unsteadiness* [98], and *intensity flicker* [26,63,77,83,84]. Figure 1.1 shows some examples of these artifacts. This figure shows frames that are corrupted by multiple artifacts. This is often the case in practice.

Not only the quality of video has been affected by time, audio tracks often suffer degradations as well. However, restoration of audio is beyond the scope of AURORA and of this thesis.

Even though a single algorithm for restoring all the artifacts at hand in an integral manner is conceivable, a modular approach was chosen to resolve the various impairments. A divide-and-conquer strategy increases the probability of (at least partial) success. Furthermore,



(a) (b) (c) (d) (e) (f)

Figure 1.1 (a,c,e) Three consecutive frames from a Charlie Chaplin film impaired by noise, blotches, and line scratches. There are also differences in intensity, which are less visible in print than on a monitor though. Zooming in on (b) noise, (d) a blotch, and (f) a scratch.

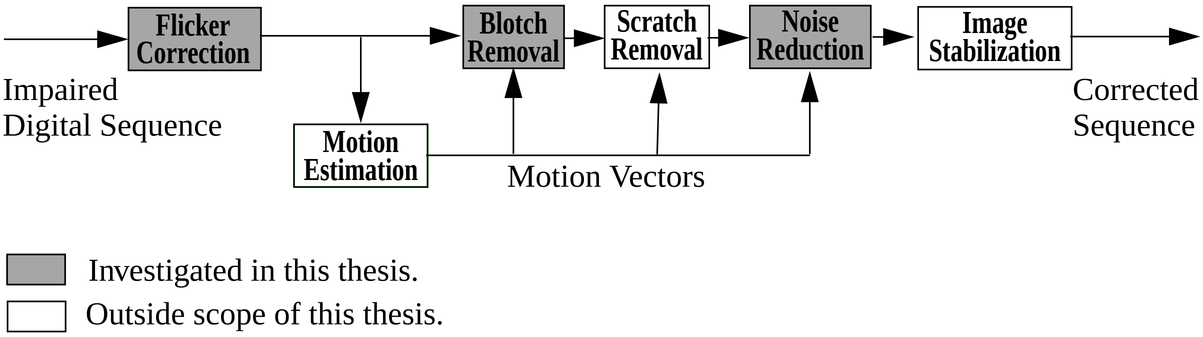


Figure 1.2 Schematic overview of a modular system towards image restoration.

real-time systems for video processing require very fast hardware for the necessary computations. Modular systems allow the computational complexity to be distributed. Figure 1.2 shows a possible system for image restoration using a modular approach that was largely implemented for the purposes of this thesis.

The first block in Figure 1.2, *flicker correction*, removes disturbing variations in image intensity in time. Intensity flicker hampers accurate local motion estimation; therefore, it is appropriate to correct this artifact prior to applying any restoration technique that relies on local motion estimates. Next, local motion is estimated. Instead of designing (yet another) motion estimator that is robust to the various artifacts, this thesis uses a hierarchical block matcher [11,31,93] with constraints on the smoothness of the motion vectors. Where the motion vectors are not reliable, due to the presence of artifacts, a strategy of vector repair is applied when necessary [16,32,47,54,65]. Next, *blotch removal* detects and removes dark and bright spots that are often visible in film sequences. *Scratch removal*, which is not a topic of research in this thesis, removes vertical line scratches. *Noise reduction* reduces the amount of noise while it preserves the underlying signal as well as possible. Finally, *image stabilization* makes the sequence steadier by aligning (registering) the frames of an image sequence in a temporal sense. Image stabilization is not a topic of research in this thesis.

In Figure 1.2, blotches and scratches are addressed prior to noise because they are local artifacts, corrections thereof influence the image contents only locally. Noise reduction is a global operation that affects each and every pixel in a frame. Therefore, all processes following noise reduction are affected by possible artifacts introduced by the noise reduction algorithm. Image stabilization, for which very robust algorithms exist, is placed at the back end of the system because it too affects each and every pixel by compensating for subpixel motion and by zooming in on the image. Zooming is required to avoid visible temporal artifacts near the image boundaries. As already mentioned, intensity flicker correction is an appropriate front end to the system. It is applied prior to the algorithms that require local motion estimates.

At the starting point in Figure 1.2 are digital image sequences instead of physical reels of film or tapes containing analog video. Rather than investigating the large number of formats and systems that have been used in one period or another over the last century, it is assumed that the archived material has been digitized by skilled technicians who know best how to digitize the film and video from the various sources. When the source material is film, digital image sequences are obtained by digitizing the output of the film-to-video *telecine*. It must be kept in mind that the earlier telecines have their limitations in terms of noise characteristics and resolution. Sometimes a copy on video tape obtained from an earlier telecine is all that remains of a film.

The output of the system in Figure 1.2 forms the restored image sequence. Subjective evaluations using test panels assess the improvement in perceived quality of the restored sequence with respect to the impaired input sequence.

1.3 Thesis outline

Chapter 2 commences with general remarks on model selection, parameter estimation, and restoration. The key to an automatic restoration system lies in automatic, reliable parameter estimation. Models for noise, blotches, line scratches, film unsteadiness, and intensity flicker are reviewed. Motion estimation is an important tool in image sequence restoration, and its accuracy determines the quality of the restored sequences. For this reason, the influence of artifacts on motion estimation is investigated. It is likely that archived material selected for preservation is re-stored in a compressed format on new digital media. To appreciate the possible benefits of image restoration with respect to compression, the influence of artifacts on the coding efficiency of encoders based on the MPEG2 video compression standard is investigated.

Chapter 3 develops a method for correcting intensity flicker. This method reduces temporal fluctuations in image intensity automatically by equalizing local image means and variances in a temporal sense. The proposed method was developed to be implemented in hardware; therefore, the number of operations per frame and the complexity of these operations have been kept as low as possible. Experimental results on artificially and naturally degraded sequences prove the effectiveness of the method.

Chapter 4 investigates blotch detection and removal. Existing methods, both heuristic and model based, are reviewed. Improved methods are developed. Specifically, the performance of a blotch detector can be increased significantly by postprocessing the detection masks resulting from this detector. The postprocessing operations take into account the influence of noise on the detection process; they also exploit the spatial coherency within blotches. Where blotches corrupt the image data, the motion estimates are not reliable. Therefore, benefits of motion-vector repair are investigated. Finally, a new, relatively fast model-based method for good-quality interpolation of missing data is presented.

Chapter 5 investigates coring. Coring is a well-known technique for removing noise from images. The mechanism of coring consists of transforming a signal into a frequency domain and reducing the transform coefficients by the coring function. The inverse transform of the cored coefficients gives the noise-reduced image. This chapter develops a framework for coring image sequences. The framework is based on 3D image decompositions, which allows temporal information to be exploited. This is preferable to processing each frame independently of the other frames in the image sequence. Furthermore, this chapter shows that coring can be imbedded into an MPEG encoder with relatively little additional complexity. The adjusted encoder significantly increases the quality of the coded noisy image sequences.

Chapter 6 evaluates the image restoration tools developed in this thesis. First, it verifies experimentally that the perceived quality of restored image sequences is better than that of the impaired source material. Second, it verifies experimentally that, for the artifacts under consideration, image restoration leads to more efficient compression. Chapter 6 concludes this thesis with a discussion.

Chapter 2

Modeling and coding

Summary. This chapter models the most common artifacts in old film and video sequences. The influence of these artifacts on motion estimation, which is an important tool for digital image sequence restoration, is investigated both qualitatively and quantitatively. Archived film and video that are preserved on new digital media are likely to be stored in compressed form. To assess the possible benefits of restoring impaired image sequences prior to encoding, the influence of artifacts on the coding efficiency is examined for the case of an MPEG2 encoder.

2.1 Modeling for image restoration

Model selection and parameter estimation are key elements in the design process of an image restoration algorithm. Section 2.1.1 reviews these key elements so that their presence can be recognized clearly in subsequent chapters. It is argued that robust automatic parameter estimation is essential to an automatic image restoration system. Section 2.1.2 models common degradations that affect old film and video sequences. These models form a basis for the restoration techniques developed in this thesis. They are also used for evaluation purposes. Section 2.1.3. investigates the influence of artifacts on the accuracy of motion estimation.

2.1.1 Model selection and parameter estimation

Image model. Many models that define various aspects of natural images and of image sequences are described in literature. For example, for still images, the magnitude of the Fourier spectrum has a *1/f* characteristic [88], and local pixel intensities depend on each other via markov random fields [28,86,104,105]; for image sequences, there is a very high correlation between frames in time for image sequences [33].

The choice of the image model to be used depends on the problem at hand. In the case of image restoration, it is appropriate to select image models with ordinary parameter values that are affected as much as possible by the degradations under investigation. The reason for this is apparent. Suppose the model parameters of the assumed image model are not affected at all by a certain degradation. Then that image model provides no information that can be used for determining the severity of that degradation, nor does it provide any indication of how to correct the degradation.

Degradation model. Degradation models describe how data are corrupted; they imply how the model parameters for unimpaired images are altered. Models for specific degradations are obtained through a thorough analysis of the mechanisms generating the artifacts. The analysis is not always straightforward because the physical processes that underlie an impairment can be very complex and difficult to qualify. Often there is a lack of detailed knowledge on how a signal was generated. In practice, approximations and assumptions that seem reasonable have to be made. For example, in Section 2.1.2, the overall influence of the various noise sources affecting pictures in a chain of image capture, conversion, and storage is approximated by a single source instead of taking into account all the individual noise contributions explicitly.

Restoration model. Ideally, restoration would be modeled as the inverse operation of the degradation with its model parameters. Unfortunately, “the inverse” does not exist in many cases due to the singularities introduced by the degradation and due to the limited accuracy with which the model parameters are known. There are many solutions to a restoration problem that give identical observed signals when the degradation model (though be it with different parameters) is applied to them. For example, image data corrupted by blotches can be restored by a number of methods (Chapter 4), each of which gives a different solution. However, none of the solutions conflict with the degradation process and with the observed data that result from the degradation process.

The restoration problem is ill posed in the sense that no unique inverse to the degradation exists. A unique solution can be found only by reducing the space of possible solutions, by setting constraints in the form of criteria that must be fulfilled as well as is possible: the characteristics of the restored image are required to fit an image model. The goal of image restoration is to restore an image so that it resembles the original scene as closely as possible. Therefore, an often used additional criterion is that, in the spatial domain, the mean squared error between the restored image and the original, uncorrupted image must be as small as possible.

Estimating model parameters. Figure 2.1 shows how the image, degradation, and restoration models relate to each other. The central element that links the models is parameter estimation (*system identification)*. The quality of a restored image sequence is determined by the quality of the estimated model parameters. Indeed, the quality of a restored image sequence can be worse than that of the degraded source material if poor choices are made for the values of the

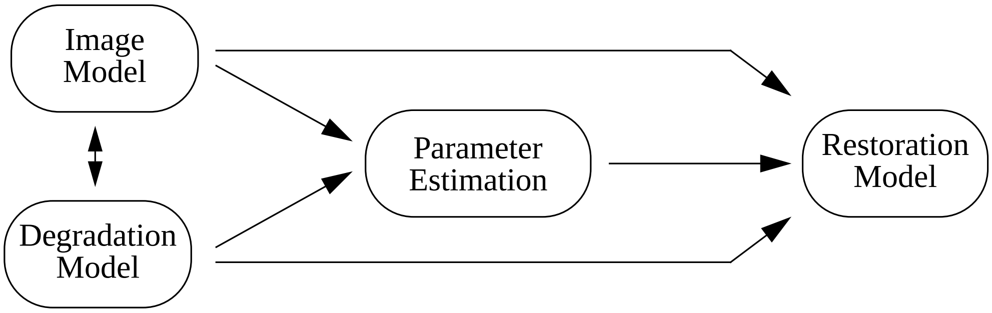


Figure 2.1 Relationships between model selection and parameter estimation.

model parameters. Therefore, the level of automation for which model parameters can be estimated in a robust manner and with sufficient accuracy determines the extent to which a restoration system performs its function without user intervention. For this reason, automatic parameter estimation from the observed signals is an important part of each of the methods for image restoration presented in this thesis.

Automatic parameter estimation is a non-trivial task in many cases due to the fact that insufficient numbers of data are available and due to the presence of noise. The term *noise* has a broad meaning in this context, and often it includes the signal to be restored from the observed data themselves. For example, estimating the noise variance (as a parameter for some algorithm) is hampered by the fact that it is very difficult to differentiate between noise and texture in natural images. Again, approximations and assumptions that seem reasonable have to be made.

Note that the quality of the estimated model parameters, e.g., determined by means of a direct numerical comparison to the true parameters, is not necessarily a good indication of the quality of the restoration result. This is because the quality of the restoration result varies in a different way for estimation errors in each of the parameters [52].

2.1.2 Impairments in old film and video sequences

Chapter 1 mentions the most common impairments in old film and video sequences, and Figure 1.1 shows some examples of these artifacts. This subsection gives models for the various impairments. Figure 2.2 indicates the sources of the artifacts in a chain of recording, storage, conversion, and digitization.

Noise. Any recorded signal is affected by noise, no matter how precise the recording apparatus. In the case of archived material, many noise sources can be pointed out. There is granular noise on film, a result of the finite size of the silver grains on film, that can be modeled by signal-dependent random processes [12,42,70,73]. There is photon or quantum noise from plumbicon tubes and *charged coupled devices* (CCDs) that is modeled as a signal-dependent Poisson process [18]. There is also thermal noise, introduced by electronic amplifiers and electronic processing, that is modeled as additive white gaussian noise [17,73]. There is impulsive noise resulting from disturbances of digital signals stored on magnetic tape [44]. Finally, in the case of digital signal processing, the digitizing process introduces quantization noise that is uniformly distributed [78].

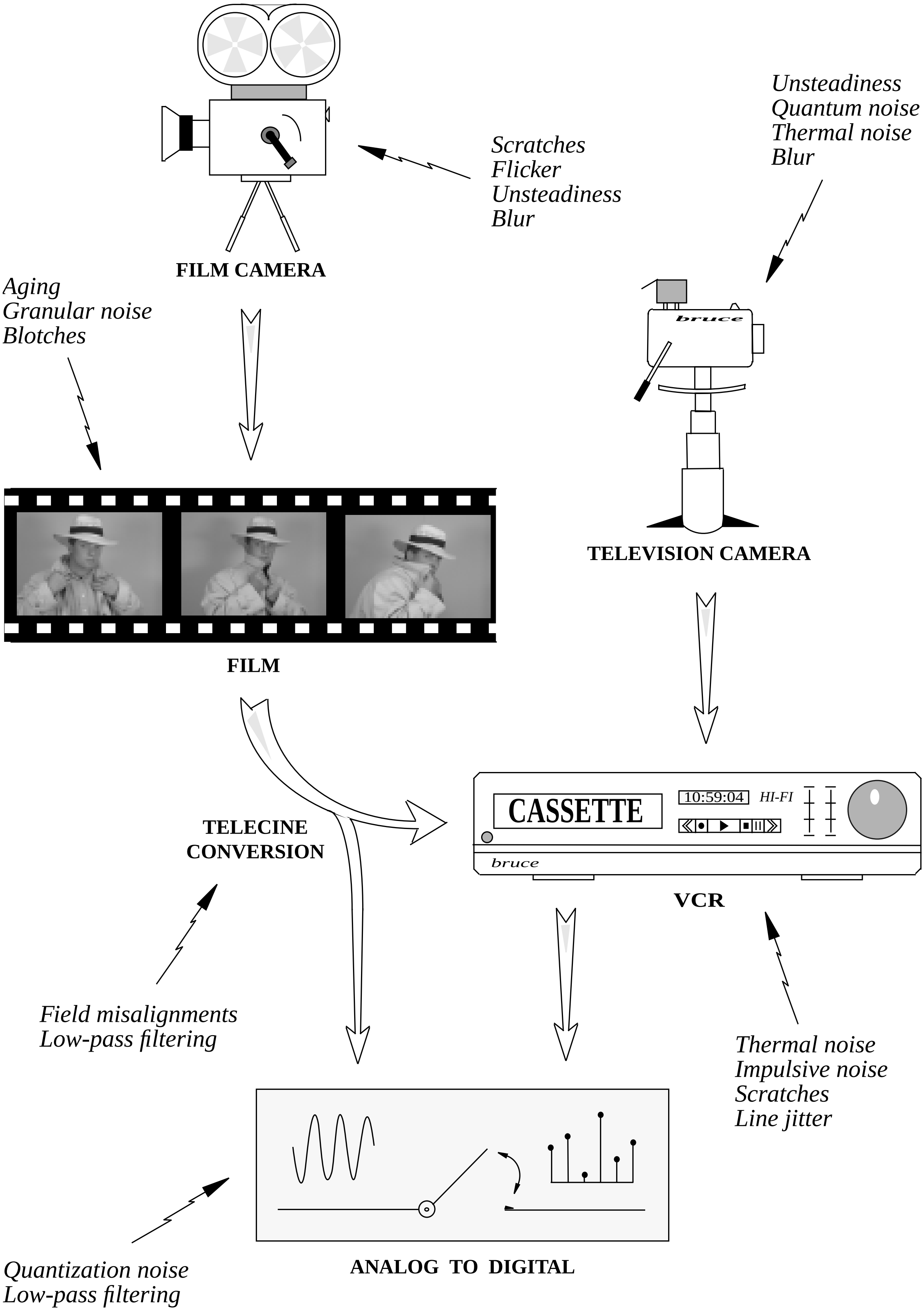


Figure 2.2 Sources of image degradation in a chain of recording, storage, conversion and digitization.

Many historical (and modern) film and video documents contain a combination of all the types of noise mentioned. For instance, such is the case for material originating on film that has been transferred to video. Modeling noise is often complicated by the band-limiting effects of optical systems in cameras and by the nonlinear gamma correction that makes the noise dependent on the signal [46]. Quantitative analysis of the contributions of each individual noise source to a recorded image is extremely difficult, if not impossible. In practice, it is often assumed that the *Central Limit Theorem* [55] applies to the various noise sources. This implies the assumption that the various noise sources generate *independent and identically distributed* (i.i.d.) noise.

Unless mentioned otherwise, this thesis assumes that the combined noise sources can be represented by a single i.i.d. additive gaussian noise source. Hence, an image corrupted by noise is modeled as follows. Let $y(\boldsymbol{i})$ with $\boldsymbol{i} = (i, j, t)$ be an image with discrete spatial coordinates (i, j) recorded at time t. Let the noise be $\eta(\boldsymbol{i})$. The observed signal $z(\boldsymbol{i})$ is then given by:

$$z(\boldsymbol{i}) = y(\boldsymbol{i}) + \eta(\boldsymbol{i}). \tag{2.1}$$

Many very different approaches to noise reduction are found in the literature, including optimal linear filtering techniques, (nonlinear) order statistics, scale-space representations, and bayesian restoration techniques [1,5,13,20,21,22,23,35,46,69,70,85].

Blotches. Blotches are artifacts that are typically related to film. In this thesis, the term *blotch* is used to indicate the effects that can result from two physical degradation processes of film. Both degradations lead to similar visual effects. The first degradation process is a result of dirt. Dirt particles covering the film introduce bright or dark spots on the picture (depending on whether the dirt is present on the negative or on the positive). The second degradation process is the loss of gelatin covering the film, which can be caused by mishandling and aging of the film. In this case, the image is said to be *blotched.* A model for blotches is given in [47]:

$$z(\boldsymbol{i}) = (1 - d(\boldsymbol{i})) \cdot y(\boldsymbol{i}) + d(\boldsymbol{i}) \cdot c(\boldsymbol{i}), \tag{2.2}$$

where $z(\boldsymbol{i})$ and $y(\boldsymbol{i})$ are the observed and the original (unimpaired) data, respectively. The binary blotch detection mask $d(\boldsymbol{i})$ indicates whether each individual pixel has been corrupted: $d(\boldsymbol{i}) \in \{0, 1\}$. The values at the corrupted sites are given by $c(\boldsymbol{i})$, with $c(\boldsymbol{i}) \neq y(\boldsymbol{i})$. A property of blotches is that the intensity values at the corrupted sites vary smoothly; that the variance $c(\boldsymbol{i})$ within a blotch is small. Blotches seldom appear at the same location in a pair of consecutive frames. Therefore the binary mask $d(\boldsymbol{i})$ will seldom be set to one at two spatially co-sited locations in a pair of consecutive frames. However, there is spatial coherence within a blotch; if a pixel is blotched, it is likely that some of its neighbors are corrupted as well.

Films corrupted by blotches are often restored in a two-step approach. The first step detects blotches and generates binary detection masks that indicate whether each pixel is part of a blotch. The second step corrects pixels by means of spatio-temporal interpolation [26,29,45,48,49,63,64,80,82,94]. Sometimes an additional step of motion estimation is included prior to interpolation because motion vectors are less reliable at corrupted sites. An alternative approach is presented in [47], where blotches are detected and corrected simultaneously.

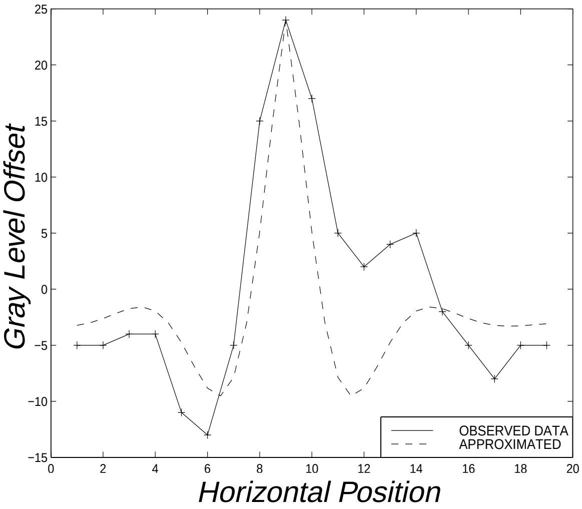


Figure 2.3 Measured intensities (solid line) and approximated intensities (dashed line) from a cross section of the vertical scratch in Figure 1.1e

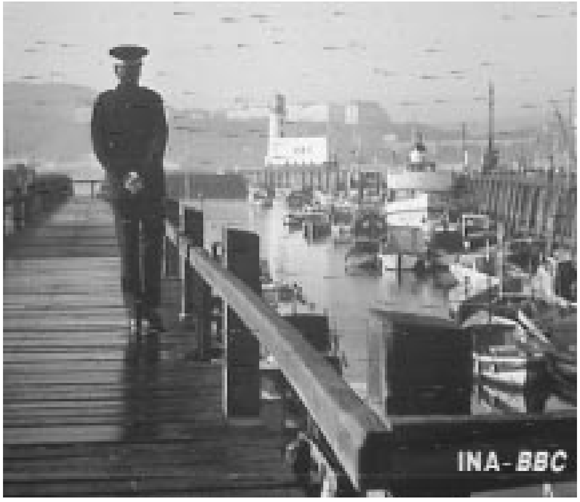


Figure 2.4 Example of a frame affected by a horizontal scratch on a two-inch video tape. (Photo by courtesy of the BBC).

Line scratches. A distinction can be made between horizontal and vertical line scratches. Vertical line scratches are impairments that are typically related to film [47,62]. They are caused by sharp particles scratching the film in a direction parallel to the direction of film transport within the camera. Line scratches are often visible as bright or dark vertical lines. The fact that vertical lines appear in nature frequently makes it difficult for an algorithm to distinguish between scratches and real-image structures. A one-dimensional cross-section of a scratch can be modeled by a damped sinusoid (Figure 2.3):

$$l(i) = A \cdot k^{|c-i|} \cdot \cos\left(\frac{|c-i|}{w}\right) + f_0, \tag{2.3}$$

where A depends on the dynamic range of the intensities over the cross-section of a scratch, k is the damping coefficient, c indicates the central position of the scratch, w indicates the width of the scratch, and f_0 is an offset determined by the local mean gray level. Once detected, line scratches can be restored by spatial or spatio-temporal interpolation.

In the case of video, horizontal scratches disturb the magnetic information stored on the tape. As a result of the helical scanning applied in video players, a horizontal scratch on the physical carrier does not necessarily give a single horizontal scratch in the demodulated image. For example, a horizontal scratch on a *two-inch* recording results in local distortions all over the demodulated image. Figure 2.4 is an example.

Film unsteadiness. Two types of film unsteadiness are defined, namely interframe and intraframe unsteadiness. The first and most important category is visible as global frame-to-frame displacements caused by mechanical tolerances in the transport system in film cameras and by unsteady fixation of the image acquisition apparatus. A model for interframe unsteadiness is:

$$z(\boldsymbol{i}) = y(i - q_i(t), j - q_j(t), t). \tag{2.4}$$

Here $q_i(t)$ and $q_j(t)$ indicate the global horizontal and vertical displacement of frame t with respect to the previous frame. Intraframe unsteadiness can be caused by transfers from film to video where the field alignment is off (many older telecines used separate optical paths for the odd and even fields). This leads to interference patterns that are perceived as variances in luminance. Unsteadiness correction is estimated from the displacements and misalignments by maximizing temporal and spatial correlation, followed by resampling of the data. See [98], for example.

Intensity flicker. Intensity flicker is defined as unnatural temporal fluctuations in the perceived image intensity that do not originate from the original scene. There are a great number of causes, e.g., aging of film, dust, chemical processing, copying, aliasing, and, in the case of the earlier film cameras, variations in shutter time. This thesis models intensity flicker as:

$$z(\boldsymbol{i}) = \alpha(\boldsymbol{i}) \cdot y(\boldsymbol{i}) + \beta(\boldsymbol{i}), \tag{2.5}$$

where fluctuations in image intensity variance and in intensity mean are represented by the multiplicative $\alpha(\boldsymbol{i})$ and additive $\beta(\boldsymbol{i})$. It is assumed that $\alpha(\boldsymbol{i})$ and $\beta(\boldsymbol{i})$ are spatially smooth functions. Histogram equalization has been proposed as a solution to intensity flicker [26,63,77]. This thesis presents a more robust solution [83].

Other artifacts are line-jitter [47,50], color fading, blur [8,52], echoes, drop-outs and moiré effects. These are beyond the scope of this thesis.

2.1.3 Influence of artifacts on motion estimation

For image sequence restoration, temporal data often provide additional information that can be exploited above that which can be extracted from spatial data only. This is because natural image sequences are highly correlated in a temporal sense in stationary regions. In nonstationary regions, object motion reduces the local temporal correlation. Therefore, increasing the stationarity of the data via motion estimation and compensation is beneficial to the restoration process. Many motion estimation techniques have been developed in the context of image compression. Examples are (hierarchical) block matchers, pel-recursive estimators, phase correlators, and estimators based on bayesian techniques [10,11,31,51,71,93].

This thesis uses a hierarchical motion estimator with integer precision and some constraints on the smoothness of the motion vectors. The constraints on smoothness are imposed by increasingly restricting the allowed deviation from the local candidate vectors passed on from lower resolution levels to higher resolution levels. Appendix A describes the details of this motion estimator. A motion-compensated frame representing a frame $y(\boldsymbol{i})$ recorded at time t computed from a reference frame recorded at time $t + k$ will be denoted as $y_{mc}(\boldsymbol{i}, t + k)$.

The scheme depicted in Figure 2.5 was used for some experiments to get some feeling for the influence of various artifacts on the accuracy of this motion estimator. In this scheme, two consecutive frames from a sequence are degraded and the motion between the objects in the

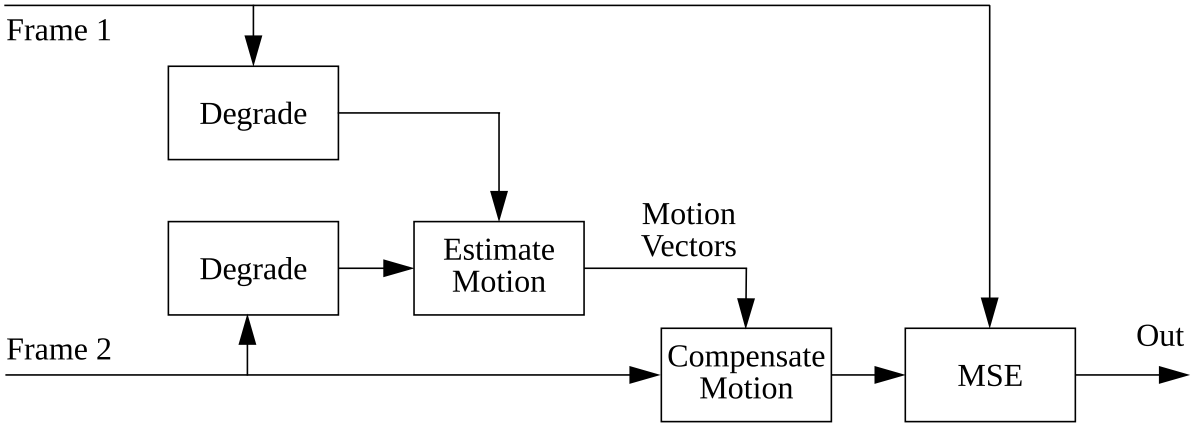


Figure 2.5 Scheme for measuring the influence of image degradations on the accuracy of estimated motion vectors.

degraded frames is estimated. The *mean squared error* (MSE) between the original (unimpaired) frames is then computed. One of these frames is compensated for motion with the estimated vectors. Let N indicate the number of pixels per frame. Then, the MSE between the current frame and the motion-compensated next frame is defined as:

$$MSE(y(\boldsymbol{i}), y_{mc}(\boldsymbol{i}, t+1)) = \frac{1}{N}\sum_{i}\sum_{j}(y(i, j, t) - y_{mc}(i, j, t, t+1))^2. \tag{2.6}$$

The rationale behind this scheme is the following. In the case that the motion estimator is not influenced much by the degradations, the correct vectors are found and the MSE is low. As the influence of the degradations on the estimated motion vectors becomes more severe, the MSE increases.

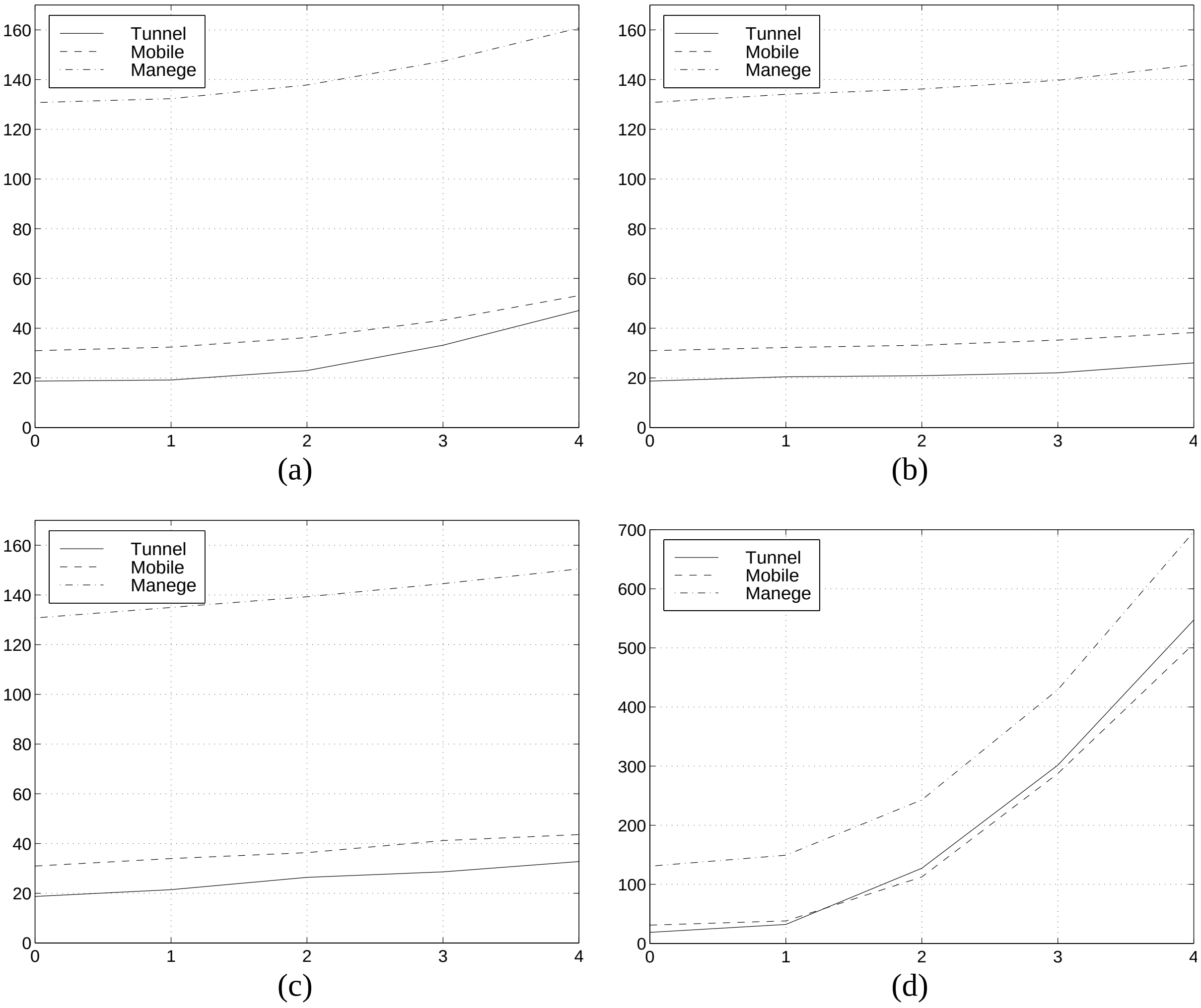
The scheme in Figure 2.5 was applied to three test sequences to which degradations of various strength are added. The first sequence, called *Tunnel*, shows a toy train driving into a tunnel. The background is steady. The second sequence, *MobCal*, has slow, subpixel motion over large image regions. The third sequence, *Manege*, shows a spinning carousel and contains a lot of motion. Table 2.1 indicates the severity of the impairments for various levels of strength. Strength zero indicates that no degradation has been added, strength four indicates an extreme level of degradation. The latter level does not occur frequently in naturally degraded image sequences.

Figure 2.6 plots the MSE for each of the test sequences as a function of the strength of the impairments. Before going into the details of the results, a few details are noted from this figure. First, in the absence of degradations, the MSE is relatively large for the *Manege* sequence. The reason for this is that the motion estimation, which was computed on a frame basis, was hampered by the strong interlacing effects. Second, the trends of the results are identical for all test sequences, i.e., the results are consistent.

Noise. Block-matching algorithms estimate motion by searching for maximal correlation between image regions in consecutive frames. If the signal-to-noise ratio is low, there is a risk

	Strength 0	Strength 1	Strength 2	Strength 3	Strength 4
Noise (variance)	0	14	56	127	225
Blotches (% corrupted)	0	0.41	0.62	1.04	1.90
Number of Scratches	0	2	5	8	11
Flicker (MSE)	0	19	72	161	281

Table 2.1 Average strength of various impairments added to test sequences. For noise the measure is the noise variance; for blotches, the measure is the percentage of pixels corrupted; for scratches, the measure is the number of scratches; and for intensity flicker, the measure is the MSE between original and corrupted frames.



(a) (b) (c) (d)

Figure 2.6 MSE versus strength of impairment (0 = no impairment, 4 = greatly impaired): (a) noise, (b) blotches, (c) line scratches, (d) intensity flicker. Note the differences in scale.

that the maximum results largely from correlating noise. In the case the noise spectrum is white, hierarchical motion estimators are more robust to noise than full-search block matchers. Most of the signal energy of natural images is concentrated in the low frequencies. For a hierarchical block matcher this means that at the lower resolution levels, which are obtained by low-pass filtering the data, the signal-to-noise ratio is higher than at the higher resolution levels. Therefore, the probability of spurious matches is reduced. The influence of noise at the higher resolution levels is reduced by the constraints placed on the smoothness of the candidate motion vectors. Figure 2.6a shows the MSE computed for the three test sequences to which various amounts of white gaussian noise have been added.

Blotches. A hierarchical block matcher will find the general direction in which data corrupted by blotches move, provided that the sizes of the contaminated areas are not too large. Because of the subsampling, the sizes of the blotches are reduced and they will have little influence on the block-matching results at the lower resolution levels. At the higher resolutions, the blotches cover larger parts of the blocks used for matching, and blotches will therefore have great influence on the matching results. However, if the number of candidate vectors is limited (e.g., in case the motion is identical in all neighboring regions) the correct motion vector may yet be found. Figure 2.6b shows the MSE computed for the three test sequences to which various numbers of blotches have been added.

Line scratches. The temporal consistency of line scratches is very good. As a result, motion estimators tend to lock onto them, especially if the contrast of the scratches is great with respect to the background. If the background motion is different from that of the line scratches, considerable errors result. Figure 2.6c shows the MSE computed for the three test sequences.

Unsteadiness. Measuring the influence of unsteadiness on motion estimates with the scheme in Figure 2.5 is not meaningful. Estimating motion between frames from an unsteady sequence is not unlike estimating motion between frames from a sequence containing camera pan. A motion estimator that performs its function well does not differentiate between global and local motion. In practice, unsteadiness (and camera pan) does have some influence. First, there are edge effects due to data moving in and out of the picture. Second, motion estimators are often intentionally biased towards zero-motion vectors. Third, the motion estimation can be influenced by aliasing if the data are not prefiltered correctly. This third effect is not of much importance because natural images have relatively little high-frequency content.

Intensity Flicker. Many motion estimators, including the hierarchical motion estimator used in this thesis, assume the *constant luminance constraint* [93]. This constraint, which requires that there be no variations in luminance between consecutive frames, is not met in the presence of intensity flicker. Figure 2.6d shows the MSE computed for the three test sequences to which varying amounts of intensity flicker have been added. The dramatic influence of this artifact on the quality of the estimated motion vectors compared to the other the artifacts examined becomes clear when the scale in Figure 2.6d is compared to those in Figures 2.6a-c.

In conclusion, artifacts can have a considerable impact on the accuracy of estimated motion vectors. In some cases, this leads to a chicken-and-egg problem: in order to obtain good motion estimates, the artifacts should be restored; and in order to restore the artifacts, good motion estimates are required. This problem can often be overcome by applying iterative solutions where estimates of the motion vectors and of the restored image are obtained in an alter-

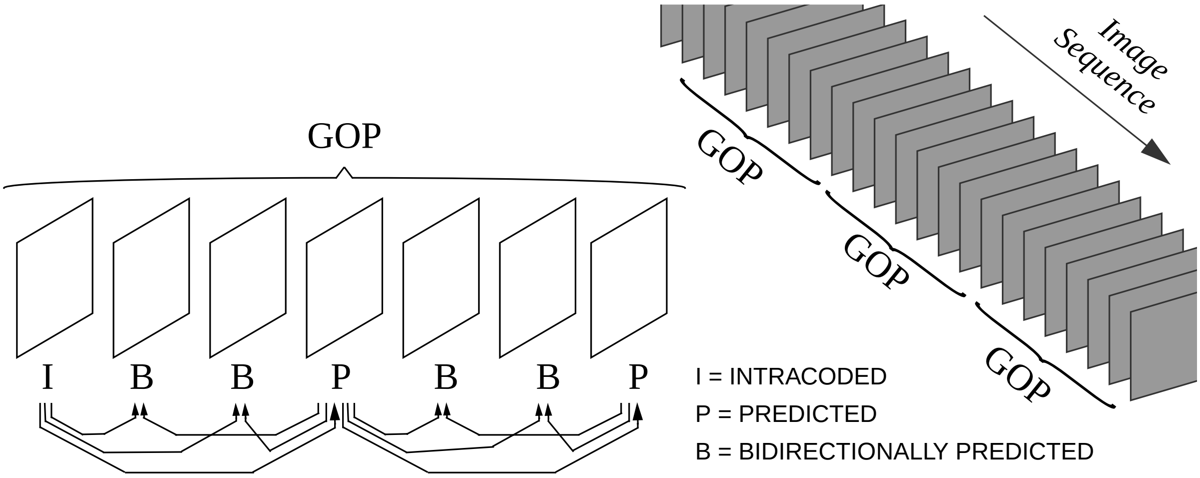


Figure 2.7 Subdivision of an image sequence into *groups of pictures* (GOPs). In this example, the GOP has length 7 and it contains I, P and B frames. The arrows indicate the prediction directions.

nating fashion. Alternatively, restoration methods that do not rely on motion estimates might be devised (Chapter 3) or a strategy of motion-vector repair can be applied after the severity of the impairments has been determined (Chapter 4).

2.2 Image restoration and storage

Restoration of archived film and video, as is the title of this thesis, implies that the restored sequences will once again be archived. It is very likely that the restored documents are stored in new digital formats rather, than in analog formats similar to those from which the material originated. Most restored material will be re-archived in a compressed form due to the high costs associated with renewed storage of the vast amounts of material being held in store currently. This section investigates the effects of various impairments on the coding efficiency and uses the MPEG2 compression standard as a reference. The results of this investigation indicate the possible benefits that can be obtained by applying image restoration prior to encoding.

2.2.1 Brief description of MPEG2

The ISO/IEC MPEG2 coding standard developed by the Motion Pictures Expert Group is currently *the* industry standard used for many digital video communication and storage applications. As a result of the requirements on its versatility, it has become a very complex standard with a description that fills several volumes [37,38,39]. The following describes only the basics of MPEG2 that are relevant to this thesis.

To achieve efficient compression, the MPEG2 encoding scheme exploits spatial and temporal redundancy within elementary units of pictures. Such an elementary unit is called a *group of pictures* (GOP) (Figure 2.7). MPEG2 defines three types of pictures that can be used within a

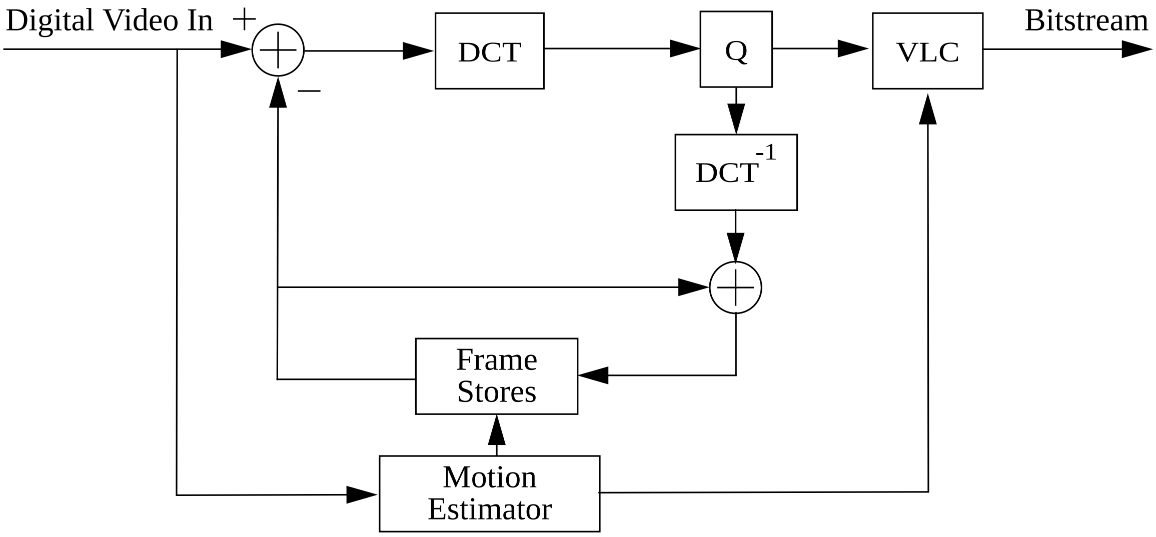


Figure 2.8 Schematic overview of the hybrid coding scheme used in MPEG2.

GOP, namely *intra frames* (I frames), *predicted frames* (P frames), and *bi-directionally interpolated frames* (B frames). A GOP cannot consist of a random collection of I, B, and P frames. There are some rules that must be adhered to, e.g., the first encoded picture in a GOP is always an I frame. Figure 2.8 gives a schematic overview of the hybrid coding scheme that forms the heart of the MPEG2 coding system.

I frames. To encode I frames, spatial information is used only. Therefore, temporal information for decoding I frames is not required. This is important because it allows random access to the image sequence (on the level of GOPs anyhow) and it limits error propagation in the temporal direction resulting from possible bit errors in a stream of encoded data.

Efficient compression of I frames requires reduction of spatial redundancy. The MPEG2 standard reduces the spatial redundancy by subdividing I frames into 8 by 8 image blocks and applying the *discrete cosine transform* (DCT) to these blocks. The decorrelating properties of the DCT concentrate much of the signal energy of natural images in the lower-frequency DCT coefficients. A quantizer Q quantizes the transform coefficients and thereby reduces the number of representation levels and sets many coefficients to zero. Note that, as the eye is less sensitive to quantization of high frequencies, the high-frequency components can be quantized relatively coarsely. Entropy coding codes the remaining coefficients efficiently by applying *run-length coding* followed by *variable length coding* (VLC) to each 8 by 8 block of quantized DCTs. The result forms the encoder output.

The decompression of I frames is straightforward: the inverse DCT is applied to 8 by 8 blocks in which the quantized coefficients are ordered after the entropy-coded data are decoded.

B and P frames. Efficient compression of P frames and B frames is achieved by exploiting both temporal and spatial redundancy. P frames are predicted from single I frames or P frames coded previously, for which motion estimation and compensation is often used. The prediction error signals, which contain spatial redundancy, are encoded as are the I frames, i.e., by means of the DCT and quantization. B frames are predicted from two coded I frames or P frames and

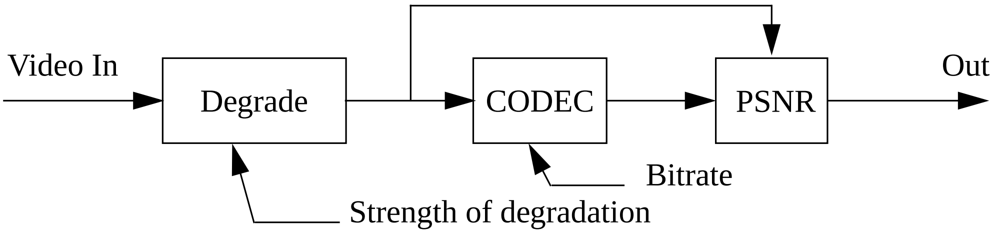


Figure 2.9 Experimental setup for evaluating the influence of artifacts on the coding efficiency.

are encoded like the P frames. The motion vectors are transmitted as well; these are encoded with differential coding. Note that, in the case of P frames and B frames, the encoder may well decide that it is more efficient to encode the original contents of an image region instead of encoding the prediction error signals.

Decompression consists of decoding the error signal and adding it to the motion-compensated prediction made in the decoder.

2.2.2 Influence of artifacts on coding efficiency

Figure 2.9 shows an experimental setup used for evaluating the quantitative influence of artifacts on the coding efficiency of an MPEG2 encoder. Coding efficiency is defined as the amount of distortion introduced by a codec under the condition of a limited bitrate, or, vice versa, as the bitrate required by a codec under condition of limited distortion. The scheme in Figure 2.9 measures the *peak-signal-to-noise-ratio* (PSNR) of a degraded image sequence after encoding and decoding $z_c(\boldsymbol{i})$. The degraded sequence prior to encoding $z_o(\boldsymbol{i})$ serves as the reference. The PSNR is defined as:

$$PSNR[z_o(\boldsymbol{i}), z_c(\boldsymbol{i})] = 10 \cdot Log\left(\frac{224^2}{\frac{1}{N}\sum_{\boldsymbol{i}}(z_o(\boldsymbol{i}) - z_c(\boldsymbol{i}))^2}\right). \qquad (2.7)$$

The numerator in (2.7) is a result of the dynamic range of the image intensities. In this thesis the allowed range of intensities is restricted to values between 16 and 240. If the degradations have little influence on the coding efficiency, the differences $z_o(\boldsymbol{i}) - z_c(\boldsymbol{i})$ will be small and the PSNR will be large. As the influence of the degradations on the coding efficiency increases, the PSNR decreases.

The degradations are introduced by applying the models for the artifacts in Section 2.1.2. Figure 2.10 plots the PSNR as a function of the bitrate of the encoder and of the strength of the impairments (Table 2.1) for the *MobCal* sequence. From this figure it can be seen that, if the strength of the impairments is held constant, the PSNR increases with increasing bitrate. This is to be expected, of course, because a signal can be encoded more accurately if more bits are

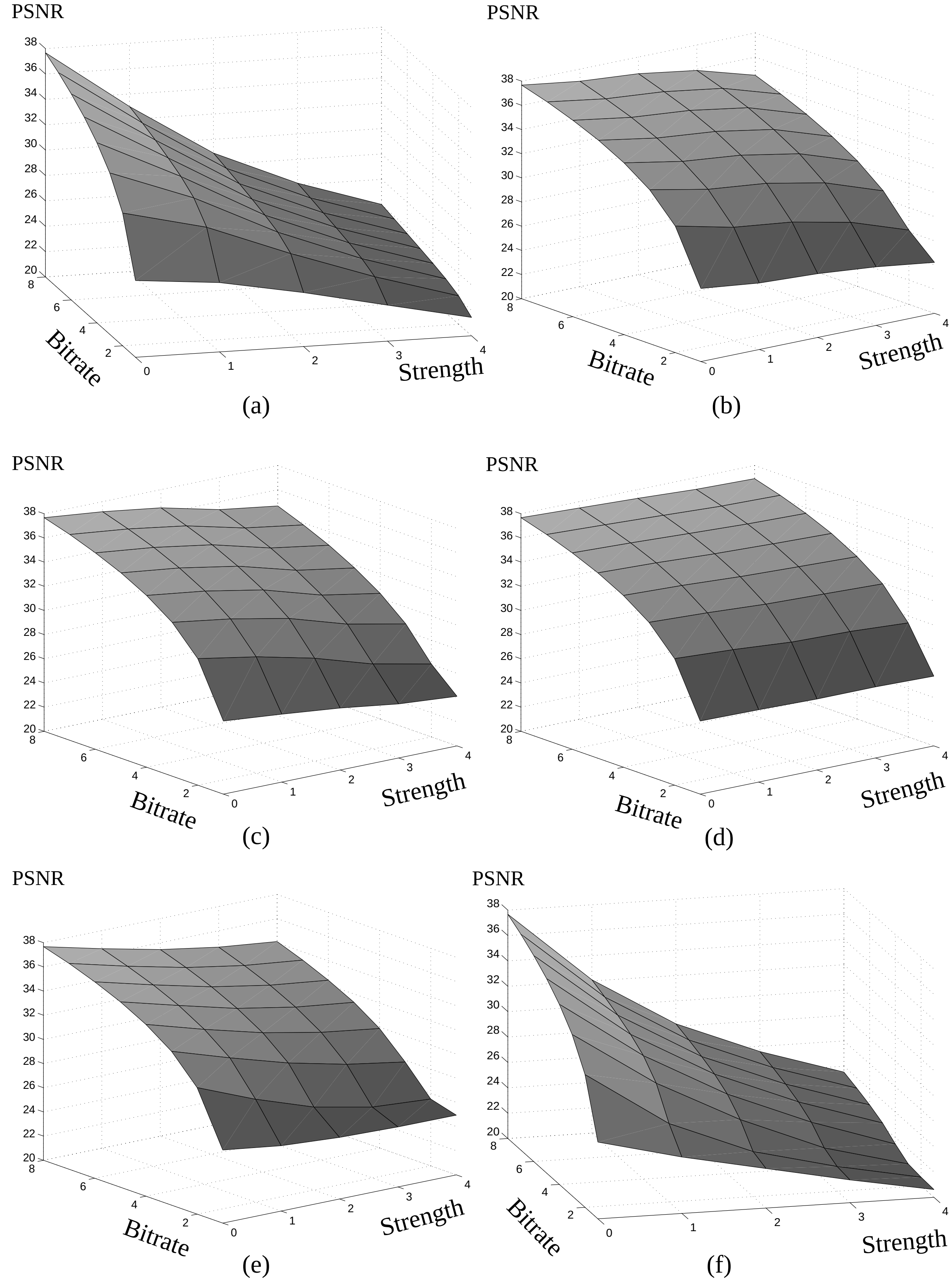


Figure 2.10 Peak-signal-to-noise ratio (PSNR) between input image and MPEG2 encoded/decoded result as a function of bitrate and strength of artifacts: (a) noise, (b) blotches, (c) line scratches, (d) unsteadiness, (e) intensity flicker and (f) all artifacts combined.

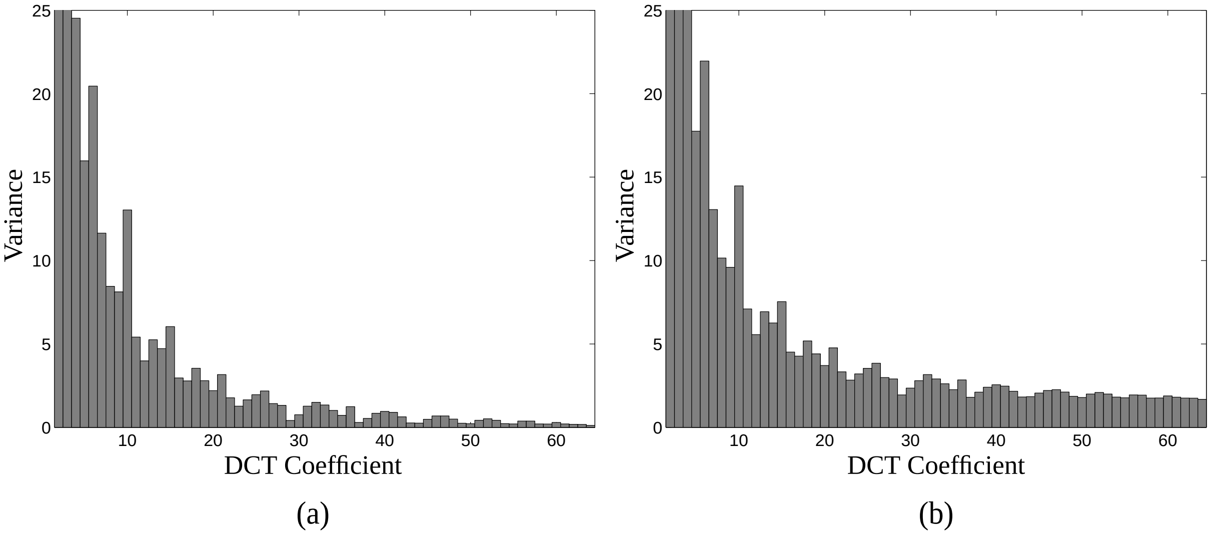


Figure 2.11 (a) Variance of DCT coefficients (in zig-zag scan order) of a clean frame from the MobCal sequence, (b) variance of DCT coefficients from same frame but now with white gaussian noise with variance 100.

available. When the bitrate is kept constant, it can been seen that the coding efficiency decreases with an increasing level of impairment. The reason for the latter is explained for each impairment in the qualitative analysis that follows.

Noise. A property of white noise is that the noise energy spreads out evenly over all the transform coefficients when an orthonormal transform is applied to it. The DCT is an orthonormal transform. Therefore, in MPEG2, the presence of additive white gaussian noise leads to fewer transform coefficients that are zero after quantization. Furthermore, on average, the amplitudes of the remaining coefficients are larger than in the noise-free case. See Figure 2.11. Both these effects lead to a decrease in coding efficiency; more coefficients must be transmitted and, on average, the codewords are longer. Similar arguments hold for the encoding of the error signals of the P frames and B frames. Note that the noise variance in the error signal is larger than that in I frames. This is so because the error signal is formed by subtracting two noisy frames. The benefits of noise reduction prior to MPEG2 encoding are shown by [77,79,81].

Blotches. Blotches replace original image contents with data that have little relation to the original scene. Large prediction errors will result for P frames and B frames at spatial locations contaminated by blotches. Large prediction errors imply nonzero DCT coefficients with large amplitudes, they therefore imply a decrease in coding efficiency. The overall influence of blotches on the coding efficiency is usually less than that of noise because blotches are local phenomena that often affect only a small percentage of the total image area.

Line scratches. Scratches are image structures that, depending on their sharpness, have high energy in the frequency domain in orientations perpendicular to that of the scratch in question. For I frames this implies nonzero coefficients with large amplitudes, i.e., a decrease in coding efficiency. The situation is slightly better for P frames and B frames if the spatial locations of the scratches do not vary too much from frame to frame. In such cases, the prediction errors are small.

Unsteadiness. In principle, the influence of film unsteadiness on prediction errors for P frames and B frames is countered by motion compensation. At first glance, the overhead due to non-zero motion vectors is neglible because of the differential coding: adjacent regions affected by global motion have only zero differential motion. However, because the codeword for no motion takes fewer bits than that for zero differential motion [25], unsteadiness influences the coding efficiency in a negative sense. Furthermore, near the image edges, the prediction errors can be large due to data moving in and out of the picture.

Intensity flicker. Intensity flicker decreases the coding efficiency of P frames and B frames for two reasons. First, the prediction error increases due to the fluctuations in image intensities. Thus the entropy of the error signal increases. Second, in the presence of intensity flicker the *constant luminance constraint* [93] under which many motion estimators operate is violated. The result is that the motion vectors are more erratic, which leads to larger differential motion. The larger the differential motion, the more bits are required for encoding. The positive effects of reducing intensity flicker prior to compression are shown by [77,79].

The analysis given here shows that artifacts have a negative influence on the coding efficiency of MPEG2. Therefore removing artifacts prior to encoding is beneficial. It is difficult to quantify the benefits beforehand because they depend strongly on the nature of the unimpaired signal, the strength of the impairments, and the effectiveness of the restoration algorithms. It should be noted that not all impairments decrease the coding efficiency. For example, image blur [8,52] is beneficial to compression because removes high frequency contents and thus nullifies the high-frequency transform coefficients.

Chapter 3

Intensity flicker correction

Summary. Intensity flicker is a common artifact in old black-and-white film sequences. It is perceived as unnatural temporal fluctuations in image intensity that do not originate from the original scene. This chapter presents a method for correcting intensity flicker on the basis of equalizing local intensity mean and variance in a temporal sense. The main problem in intensity flicker correction is concerned with estimating the model parameters in a robust fashion. This is easy enough for temporally stationary sequences. However, where local object motion hampers the estimation process measures have to be taken to avoid incorrect parameter estimates. Therefore, a motion detector that is robust to intensity flicker is developed. Where motion is detected, the model parameters are spatially interpolated from model parameters computed from stationary regions. The intensity-flicker correction system has been applied successfully to artificially and naturally degraded image sequences.

3.1 Introduction

Intensity flicker is a common artifact in old black-and-white film sequences. It is perceived as unnatural temporal fluctuations in image intensity that do not originate from the original scene. Intensity flicker has a great number of causes, e.g., aging of film, dust, chemical processing, copying, aliasing, and, in the case of the earlier film cameras, variations in shutter time. Neither equalizing the intensity histograms nor equalizing the mean frame values of consecutive frames, as suggested in [26,63,77], are general solutions to the problem. These methods do not take changes in scene contents into account, and they do not appreciate the fact that intensity flicker can be a spatially localized effect. This chapter describes a method for equalizing local intensity means and variances in a temporal sense to reduce the undesirable temporal fluctuations in image intensities [83].

Section 3.2 models the effects of intensity flicker, and derives a solution to this problem for stationary sequences that is robust to the wide range of causes of this artifact. The derived solution is optimal in a *linear mean square error* sense. The sensitivity to errors in estimated model parameters and the reliability of those parameters are analyzed. Section 3.3 extends the applicability of the method to include nonstationary sequences by incorporating motion. In the presence of intensity flicker, it is difficult to compensate for motion of local objects in order to satisfy the requirement of temporal stationarity. A strategy of compensating for global motion (camera pan) in combination with a method for detecting the remaining local object motion is applied. The model parameters are interpolated where local motion is detected. Section 3.4 shows the overall system of intensity-flicker correction and discusses some practical aspects. Section 3.5 describes experiments and results. This chapter concludes with a discussion.

3.2 Estimating and correcting intensity flicker in stationary sequences

3.2.1 A model for intensity flicker

It is not practical to find explicit physical models for each of the mechanisms mentioned that cause intensity flicker. Instead, the approach taken here models the effects of this phenomenon on the basis of the observation that intensity flicker causes temporal fluctuations in local intensity mean and variance. Since noise is unavoidable in the various phases of digital image formation, a noise term is included in the model:

$$z(\boldsymbol{i}) = \alpha(\boldsymbol{i}) \cdot y(\boldsymbol{i}) + \beta(\boldsymbol{i}) + \eta(\boldsymbol{i}). \tag{3.1}$$

The multiplicative and additive intensity-flicker parameters are denoted by $\alpha(\boldsymbol{i})$ and $\beta(\boldsymbol{i})$. In the ideal case, when no intensity flicker is present, $\alpha(\boldsymbol{i}) = 1$ and $\beta(\boldsymbol{i}) = 0$ for all $\boldsymbol{i}$. It is assumed that $\alpha(\boldsymbol{i})$ and $\beta(\boldsymbol{i})$ are spatially smooth functions. Note that $y(\boldsymbol{i})$ does not necessarily need to represent the original scene intensities; it may represent a signal that, prior to the introduction of intensity flicker, may already have been distorted. The distortion could be due to signal-dependent additive granular noise that is characteristic of film [12,70], for example.

The intensity-flicker-independent noise, denoted by $\eta(\boldsymbol{i})$, models the noise that has been added to the signal after the introduction of intensity flicker. It is assumed that this noise term is uncorrelated with the original image intensities. It is also assumed that $\eta(\boldsymbol{i})$ is a zero-mean signal with known variance. Examples are quantization noise and thermal noise originating from electronic studio equipment (VCR, amplifiers, etc.).

Correcting intensity flicker means estimating the original intensity for each pixel from the observed intensities. Based on the degradation model in (3.1), the following choice for a linear estimator for estimating $y(\boldsymbol{i})$ is obvious:

$$\hat{y}(\boldsymbol{i}) = a(\boldsymbol{i}) \cdot z(\boldsymbol{i}) + b(\boldsymbol{i}). \tag{3.2}$$

If the error between the original image intensity and the estimated original image intensity is defined as:

$$\varepsilon(\boldsymbol{i}) = y(\boldsymbol{i}) - \hat{y}(\boldsymbol{i}). \tag{3.3}$$

then it can easily be determined that, given $\alpha(\boldsymbol{i})$ and $\beta(\boldsymbol{i})$, the optimal values for $a(\boldsymbol{i})$ and $b(\boldsymbol{i})$ in a *linear minimum mean square error* (LMMSE) sense are given by:

$$a(\boldsymbol{i}) = \frac{var[z(\boldsymbol{i})] - var[\eta(\boldsymbol{i})]}{var[z(\boldsymbol{i})]} \cdot \frac{1}{\alpha(\boldsymbol{i})}, \tag{3.4}$$

$$b(\boldsymbol{i}) = -\frac{\beta(\boldsymbol{i})}{\alpha(\boldsymbol{i})} + \frac{var[\eta(\boldsymbol{i})]}{var[z(\boldsymbol{i})]} \cdot \frac{E[z(\boldsymbol{i})]}{\alpha(\boldsymbol{i})}, \tag{3.5}$$

where $E[\cdot]$ stands for the expectation operator and $var[\cdot]$ indicates the variance. It is interesting that it follows from (3.4) and (3.5) that $a(\boldsymbol{i}) = 1/\alpha(\boldsymbol{i})$ and $b(\boldsymbol{i}) = -\beta(\boldsymbol{i})/\alpha(\boldsymbol{i})$ in the absence of noise. In such a case, it follows from (3.1) and (3.2) that $\hat{y}(\boldsymbol{i}) = y(\boldsymbol{i})$. That is to say, the estimated intensities are exactly equal to the original intensities. In the extreme case that the observed signal variance equals the noise variance, we find that $a(\boldsymbol{i}) = 0$ and $\hat{y}(\boldsymbol{i}) = b(\boldsymbol{i}) = E[y(\boldsymbol{i})]$; the estimated intensities equal the expected values of the original intensities.

In practical situations, the true values for $\alpha(\boldsymbol{i})$ and $\beta(\boldsymbol{i})$ are not known and estimates $\hat{\alpha}(\boldsymbol{i})$ and $\hat{\beta}(\boldsymbol{i})$ are made from the observed data (this is the topic of Section 3.2.2). Because these estimates will never be perfect, the effects of errors in $\hat{\alpha}(\boldsymbol{i})$ and $\hat{\beta}(\boldsymbol{i})$ on $\hat{y}(\boldsymbol{i})$ is investigated. To simplify the analysis, the influence of noise is discarded. For ease of notation, the following analysis leaves out the spatial and temporal indices. Let $\hat{\alpha} = \alpha + \Delta\alpha$ and $\hat{\beta} = \beta + \Delta\beta$. The reconstruction error Δy is then given by:

$$\begin{aligned} \Delta y &= y - \hat{y} \\ &= \frac{\Delta\alpha}{\alpha + \Delta\alpha} \cdot y + \frac{\Delta\beta}{\alpha + \Delta\alpha}. \end{aligned} \tag{3.6}$$

Figure 3.1 plots the reconstruction error as a function of $\Delta\alpha$ and $\Delta\beta$ with $\alpha = 1$, $\beta = 0$ and $y = 100$. Now, if $|\Delta\alpha| \ll \alpha$, then it can be seen that the sensitivity of Δy to errors in $\hat{\alpha}(\boldsymbol{i})$ is linear in y, and that the sensitivity of Δy to errors in $\hat{\beta}(\boldsymbol{i})$ is constant:

$$\frac{d\Delta y}{d\Delta\alpha} = y \text{ and } \frac{d\Delta y}{d\Delta\beta} = 1. \tag{3.7}$$

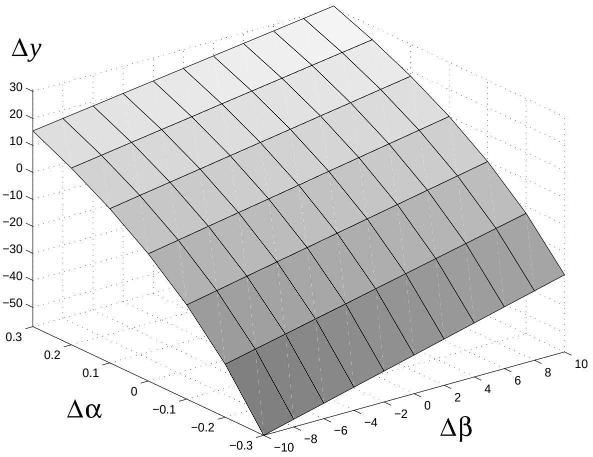


Figure 3.1 Error Δy in a reconstructed image as a function of errors $\Delta\alpha$ and $\Delta\beta$ computed for $y = 100$.

Equation (3.7) shows that Δy is much more sensitive to errors in $\hat{\alpha}(\boldsymbol{i})$ than to errors in $\hat{\beta}(\boldsymbol{i})$. It also shows that the sensitivity due to errors $\Delta\alpha$ can be minimized in absolute terms by centering the range of image intensities around 0. For example, consider a noiseless case in which $\hat{\alpha} = \alpha + 0.1$ and $\hat{\beta} = \beta$. If y ranges between 0 and 255 with $\alpha = 1$ and $\beta = 0$, then it can be seen from (3.6) that Δy is maximally 23.2. After the range of image intensities is centered around 0, y ranges between -127 and 128. The maximal absolute error is halved and, unlike the previous case, the sensitivity to errors in $\hat{\alpha}(\boldsymbol{i})$ for the mid-gray values is relatively small.

3.2.2 Estimating intensity-flicker parameters in stationary scenes

In the previous section, a LMMSE solution to intensity flicker is derived on the assumption that the intensity-flicker parameters $\alpha(\boldsymbol{i})$ and $\beta(\boldsymbol{i})$ are known. This is not the case in most practical situations, and these parameters will have to be estimated from the observed data. This section determines how the intensity-flicker parameters can be estimated from temporally stationary image sequences, i.e., image sequences that do not contain motion. It was already assumed that $\alpha(\boldsymbol{i})$ and $\beta(\boldsymbol{i})$ are spatially smooth functions. For practical purposes it is now also assumed that the intensity-flicker parameters are constant locally:

$$\begin{cases} \alpha(i, j, t) = \alpha_{m,n}(t) \\ \\ \beta(i, j, t) = \beta_{m,n}(t) \end{cases} \qquad \forall\ i, j \in \Omega_{m,n}, \tag{3.8}$$

where $\Omega_{m,n}$ indicates a small image region. The image regions $\Omega_{m,n}$ can, in principle, have any shape, but they are rectangular blocks in practice, and m, n indicate their horizontal and vertical spatial locations. The $\alpha_{m,n}(t)$ and $\beta_{m,n}(t)$ correspondig to $\Omega_{m,n}$ are considered frame-dependent matrix entries at m, n. The size $M \times N$ of the matrix depends on the total number of blocks in the horizontal and vertical directions.

Keep in mind the assumption that the zero-mean noise $\eta(\boldsymbol{i})$ is signal independent. The expected value and variance of $z(\boldsymbol{i})$ taken from (3.1) in a spatial sense for $i, j \in \Omega_{m,n}$ is given by:

$$E[z(\boldsymbol{i})] = \alpha_{m,n}(t) \cdot E[y(\boldsymbol{i})] + \beta_{m,n}(t), \tag{3.9}$$

$$var[z(\boldsymbol{i})] = \alpha^2_{m,n}(t) \cdot var[y(\boldsymbol{i})] + var[\eta(\boldsymbol{i})]. \tag{3.10}$$

Rewriting (3.9) and (3.10) gives exact analytical expressions for $\alpha_{m,n}(t)$ and $\beta_{m,n}(t)$ for $i, j \in \Omega_{m,n}$:

$$\beta_{m,n}(t) = E[z(\boldsymbol{i})] - \alpha_{m,n}(t) \cdot E[y(\boldsymbol{i})], \tag{3.11}$$

$$\alpha_{m,n}(t) = \sqrt{\frac{var[z(\boldsymbol{i})] - var[\eta(\boldsymbol{i})]}{var[y(\boldsymbol{i})]}}. \tag{3.12}$$

Equations (3.11) and (3.12) must now be solved in a practical situation. The means and variances of $z(\boldsymbol{i})$ can be estimated directly from the observed data of regions $\Omega_{m,n}$. The noise variance is assumed to be known or estimated. What remains to be estimated are the expected values and variances of $y(\boldsymbol{i})$ in the various regions $\Omega_{m,n}$.

Two methods for estimating the mean and variance of $y(\boldsymbol{i})$ for $i, j \in \Omega_{m,n}$ are discussed here. The first method estimates $y(\boldsymbol{i})$ by averaging the observed data in a temporal sense. In this case the underlying assumption is that the effects of flicker will be averaged out:

$$E[y(i,j,t)] = \frac{1}{p+q+1} \cdot \sum_{l=-p}^{q} E[z(i,j,t+l)], \tag{3.13}$$

$$var[y(i,j,t)] = \frac{1}{p+q+1} \cdot \sum_{l=-p}^{q} var[z(i,j,t+l)]. \tag{3.14}$$

The second method takes the frame corrected previously as a reference:

$$E[y(i,j,t)] = E[\hat{y}(i,j,t-1)], \tag{3.15}$$

$$var[y(i,j,t)] = var[\hat{y}(i,j,t-1)]. \tag{3.16}$$

The latter approach is adopted here because it has the significant advantage that the requirement of temporal stationarity is more likely to be fulfilled when a single reference frame, rather than multiple reference frames, is used. This approach is also more attractive in terms of computational load and memory requirements. Hence, for $i, j \in \Omega_{m,n}$, the estimated intensity-flicker parameters are given by:

$$\hat{\beta}_{m,n}(t) = E[z(i,j,t)] - \hat{\alpha}_{m,n}(t) \cdot E[\hat{y}(i,j,t-1)], \tag{3.17}$$

$$\hat{\alpha}_{m,n}(t) = \sqrt{\frac{var[z(i,j,t)] - var[\eta(i,j,t)]}{var[\hat{y}(i,j,t-1)]}}. \tag{3.18}$$

3.2.3 Measure of reliability for the estimated model parameters

Note that, by using (3.15) and (3.16), recursion is introduced into the method for flicker correction. As a result, there is a risk of error propagation leading to considerable distortions in a corrected sequence. A source of errors lies in the estimated model parameters $\hat{\alpha}_{m,n}(t)$ and $\hat{\beta}_{m,n}(t)$, which may not be exact. Therefore, it is useful to have a measure of reliability for $\hat{\alpha}_{m,n}(t)$ and $\hat{\beta}_{m,n}(t)$ that can be used to control the correction process by means of weighting and smoothing the estimated model parameters as is done in Section 3.3.3.

The $\hat{\alpha}_{m,n}(t)$ and $\hat{\beta}_{m,n}(t)$ are not very reliable in a number of cases. The first case is that of uniform image intensities. For any original image intensity in a uniform region, there are infinite combinations of $\alpha(\boldsymbol{i})$ and $\beta(\boldsymbol{i})$ that lead to the same observed intensity. The second case in which $\hat{\alpha}_{m,n}(t)$ and $\hat{\beta}_{m,n}(t)$ are potentially unreliable is caused by the fact that (3.15) and (3.16) discard the noise in $\hat{y}(\boldsymbol{i})$ originating from $\eta(\boldsymbol{i})$. This leads to values for $\hat{\alpha}_{m,n}(t)$ that are too small. Considerable errors result in regions $\Omega_{m,n}$ in which the signal variance is smaller than the noise variance.

The signal-to-noise ratio, defined as $var(y)/var(\eta)$, determines the variance of the errors in the estimated model parameters. Figure 3.2 illustrates this by plotting the reciprocal values of the error variances $\sigma^2_{\Delta\alpha}$ and $\sigma^2_{\Delta\beta}$ as a function of signal-to-noise ratio. These values were obtained experimentally by synthesizing 100 000 textured areas of 30×30 pixels with a 2D autoregressive model to which gaussian noise and flicker were added. The flicker parameters were then determined with (3.11) and (3.12). Figure 3.2 shows that the variance in the estimated model parameters is inversely proportional to the signal-to-noise ratio.

In Section 3.3.3, the model parameters that are estimated over an image are smoothed and weighted using a 2D polynomial fit. The weighted least-squares estimate of the polynomial coefficients is optimal if the weights are proportional to $1/\sigma_{\Delta\alpha}$ and $1/\sigma_{\Delta\beta}$ [92], i.e., if the weights are proportional to the squared root of the signal-to-noise ratio. Hence, the following measure of reliability $W_{m,n}(t)$, for $i, j \in \Omega_{m,n}$, is defined:

$$W_{m,n}(t) = \begin{cases} 0 & \forall \; var[z(\boldsymbol{i})] < T_n \\ \sqrt{\dfrac{var[z(\boldsymbol{i})] - T_n}{T_n}} & \text{otherwise} \end{cases}, \tag{3.19}$$

where T_n is a threshold depending on the variance of $\eta(\boldsymbol{i})$. Large values for $W_{m,n}(t)$ indicate reliable estimates; small values indicate unreliable estimates.

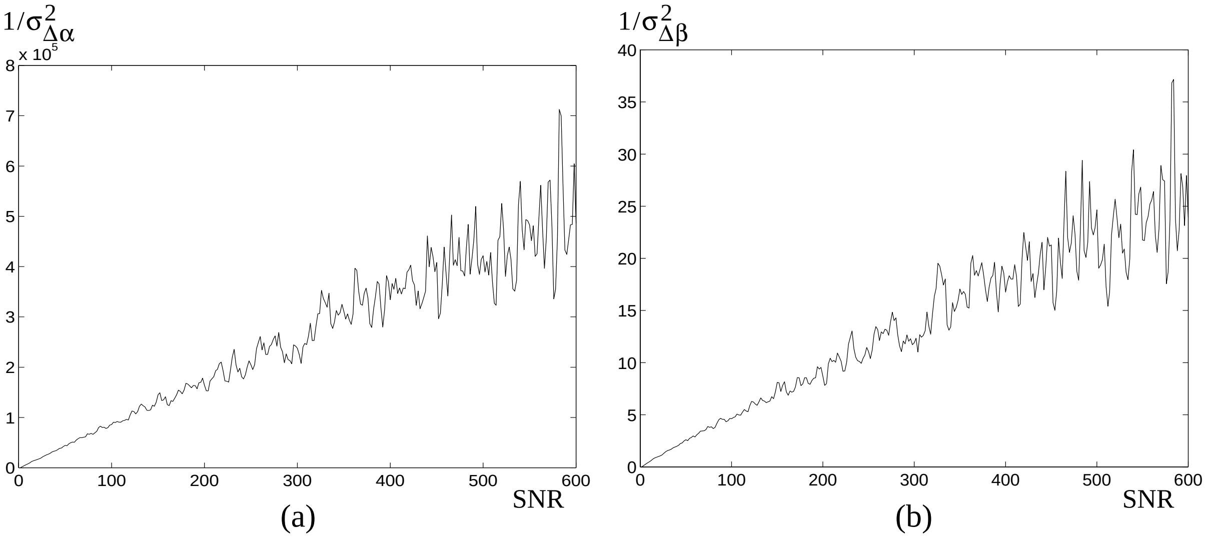


Figure 3.2 (a) Plot of $1/\sigma^2_{\Delta\alpha}$ vs. signal-to-noise ratio, (b) plot $1/\sigma^2_{\Delta\beta}$ vs. signal-to-noise ratio. Note that the relationships are linear.

3.3 Incorporating motion

The previous sections model the effects of intensity flicker and derive a solution for temporally stationary sequences. The necessity of temporal stationarity is reflected by (3.15) and (3.16), which assume that the mean and variance of $\hat{y}(i, j, t)$ and $\hat{y}(i, j, t-1)$ are identical. Real sequences, of course, are seldom temporally stationary. Measures will have to be taken to avoid estimates of $\alpha(\boldsymbol{i})$ and $\beta(\boldsymbol{i})$ that are incorrect due to motion. Compensating motion between $z(i, j, t)$ and $\hat{y}(i, j, t-1)$ helps satisfy the assumption of temporal stationarity. This requires motion estimation.

Robust methods for estimating global motion (camera pan) that are relatively insensitive to fluctuations in image intensities exist. Unfortunately, the presence of intensity flicker hampers the estimation of local motion (motion in small image regions) because local motion estimators usually have a constant luminance constraint. This includes pel-recursive methods and all motion estimators that make use of block matching in one stage or another [93]. Even if motion can be well compensated, a strategy is required for correcting flicker in previously occluded regions that have become uncovered.

For these reasons, the strategy presented here for estimating the intensity-flicker parameters in temporally nonstationary scenes is based on local motion detection. First, a pair of frames are registered to compensate for global motion (Section 3.3.1). Then the intensity-flicker parameters are estimated as outlined in Section 3.2.2. With these parameters, the remaining local motions is detected (Section 3.3.2). Finally, the missing model parameters in the temporally nonstationary regions are spatially interpolated from surrounding regions without local motion (Section 3.3.3).

3.3.1 Estimating global motion with phase correlation

In sequences with camera pan, applying global motion compensation helps satisfy the requirement of stationarity. Let the global displacement vector be $(q_i, q_j)^T$. Global motion compensation can be applied to the model parameter estimation by replacing (3.17) and (3.18) with:

$$\hat{\beta}_{m,n}(t) = E[z(i,j,t)] - \hat{\alpha}_{m,n}(t) E[\hat{y}(i-q_i, j-q_j, t-1)], \tag{3.20}$$

$$\hat{\alpha}_{m,n}(t) = \sqrt{\frac{var[z(i,j,t)] - var[\eta(i,j,t)]}{var[\hat{y}(i-q_i, j-q_j, t-1)]}}. \tag{3.21}$$

Global motion compensation is only useful if the global motion vectors (one vector to each frame) are accurate: i.e., if the global motion estimator is robust against intensity flicker. A global motion estimator that meets this requirement is one that is based on the phase correlation method applied to high-pass-filtered versions of the images [71,93].

The phase correlation method estimates motion by measuring phase shifts in the Fourier domain. This method is relatively insensitive to fluctuations in image intensity because it uses Fourier coefficients that are normalized by their magnitude. The direction of changes in intensity over edges and textured regions is preserved in the presence of intensity flicker because the amount of intensity flicker was assumed to vary smoothly in a spatial sense. This means that the phases of the higher-frequency components will not be affected by intensity flicker. However, the local mean intensities can vary considerably from frame to frame, and this gives rise to random variations in the phase of the low-frequency components. These random variations are disturbing factors in the motion estimation process that can be avoided by removing the low-pass frequency components from the input images.

The phase correlation technique estimates phase shifts in the Fourier domain as follows:

$$C_{t,t-1}(\omega_1, \omega_2) = \frac{Z_t(\omega_1, \omega_2) \cdot Z^*_{t-1}(\omega_1, \omega_2)}{\left\| Z_t(\omega_1, \omega_2) \cdot Z^*_{t-1}(\omega_1, \omega_2) \right\|}, \tag{3.22}$$

where $Z_t(\omega_1, \omega_2)$ stands for the 2D Fourier transform of $z(i,j,t)$, and * denotes the complex conjugate. If $z(i,j,t)$ and $z(i,j,t-1)$ are spatially shifted, but otherwise identical images, the inverse transform of (3.22) produces a delta pulse in the 2D correlation function. Its location yields the global displacement vector $(q_i, q_j)^T$.

3.3.2 Detecting the remaining local motion

It is important to detect the remaining local motion after compensating for global motion. Local motion causes changes in local image statistics that are not due to intensity flicker. This leads to incorrect estimates of $\alpha(\boldsymbol{i})$ and $\beta(\boldsymbol{i})$; to visible artifacts in the corrected image

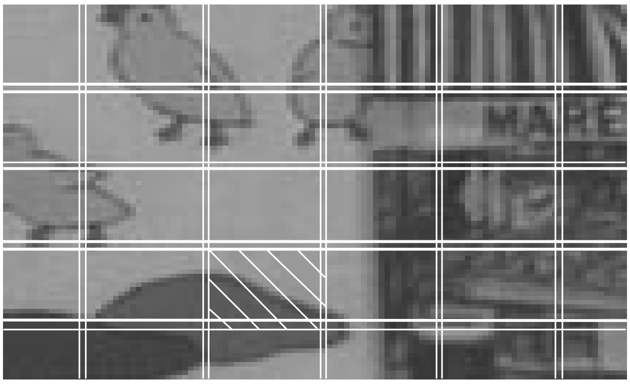


Figure 3.3 Example of part of a frame subdivided in blocks $\Omega_{m,n}$ that overlap each other by one pixel.

sequence. First, two obvious approaches to motion detection are discussed. It is concluded that these are not appropriate. Next, a robust alternative strategy is described.

Two methods for detecting local motion are (1) detecting large local frame differences between the corrected current and previous frames and (2) comparing the estimated intensity-flicker parameters $\hat{\alpha}_{m,n}(t)$ and $\hat{\beta}_{m,n}(t)$ to threshold values and detect motion when these thresholds are exceeded. These methods have disadvantages that limit their usefulness. The first method is very sensitive to film unsteadiness; slight movements of textured areas and edges lead to large frame differences and thus to “false” detections of motion. The second method requires threshold values that detect motion accurately without generating too many false alarms. Good thresholds are difficult to find because they depend on the amount of intensity flicker and the amount of local motion in the sequence.

To overcome problems resulting from small motion and hard thresholds, a robust motion-detection algorithm that relies on the current frame only is developed here. The underlying assumption of the method is that motion should only be detected if visible artifacts would otherwise be introduced. First, the observed image is subdivided into blocks $\Omega_{m,n}$ that overlap their neighbors both horizontally and vertically (Figure 3.3). The overlapping boundary regions form sets of reference intensities. The intensity-flicker parameters are estimated for each block by (3.20) and (3.21). These parameters are used with (3.2), (3.4), and (3.5) for correcting the intensities in the boundary regions. Then, for each pair of overlapping blocks, the common pixels that are assigned significantly different values are counted:

$$n_{q,r} = \sum_{\boldsymbol{i} \in S_{q,r}} boolean[|\hat{y}_q(\boldsymbol{i}) - \hat{y}_r(\boldsymbol{i})| > T_d]. \tag{3.23}$$

Here q and r indicate two adjacent image blocks, $S_{q,r}$ indicates the set of boundary pixels, T_d is a threshold above which pixels are considered to be significantly different and $boolean[\cdot]$ is a boolean function that is one if its argument is true and is zero otherwise. Motion is flagged in both regions q and r if too many pixels are significantly different, that is, if:

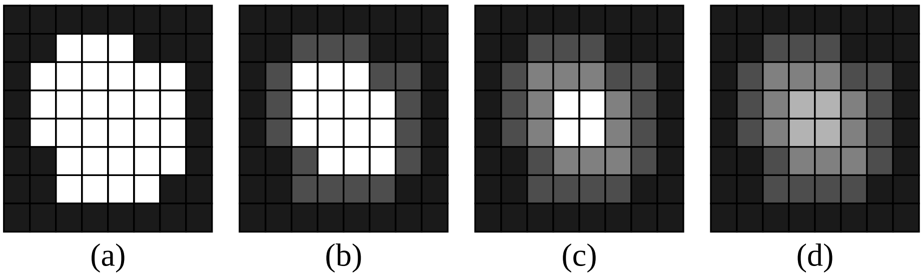


Figure 3.4 Interpolation process using dilation: (a) initial situation, (b), (c), (d) results after 1, 2 and 3 iterations.

$$n_{q,r} > D_{max}, \tag{3.24}$$

where D_{max} is a constant.

3.3.3 Interpolating missing parameters

Due to noise and motion, the estimated intensity-flicker parameters are unreliable in some cases. These parameters are referred to as *missing*. The other parameters are referred to as *known*. The goal is to find estimates of the missing parameters by means of interpolation. It is also necessary to smooth the known parameters to avoid sudden changes in local intensity in the corrected sequence. The interpolation and smoothing functions should meet the following requirements. First, the system of intensity-flicker correction should switch itself off when the correctness of the interpolated values is less certain. This means that the interpolator should incorporate biases for $\hat{\alpha}_{m,n}(t)$ and $\hat{\beta}_{m,n}(t)$ towards unity and zero, respectively, that grow as the smallest distance to a region with known parameters becomes larger. Second, the reliability of the known parameters should be taken into account.

Three methods that meet these requirements are investigated. Each of these methods uses the $W_{m,n}(t)$ determined by the measure of reliability as defined in (3.19). The interpolation and smoothing algorithms are described for the case of the multiplicative parameters $\hat{\alpha}_{m,n}(t)$. The procedures for the $\hat{\beta}_{m,n}(t)$ are similar and are not described here.

Interpolation by dilation. With each iteration of this iterative dilation approach, regions of known parameters grow at the boundaries of regions with missing parameters. Consider the matrix containing the known $\hat{\alpha}_{m,n}(t)$ corresponding to the regions $\Omega_{m,n}$ for a frame t. Figure 3.4a graphically depicts such a matrix that can be divided into two areas: the black region indicates the matrix entries for which the multiplicative parameters are known, and the white region indicates the missing entries. Each missing $\hat{\alpha}_{m,n}(t)$ and its corresponding weight $W_{m,n}(t)$ at the boundary of the two regions is interpolated by:

$$\hat{\alpha}_{m,n}(t) = \frac{\sum_{\{q,r\} \in S_{m,n}} W_{q,r}(t) \cdot \hat{\alpha}_{q,r}(t)}{\sum_{\{q,r\} \in S_{m,n}} W_{q,r}(t)} \cdot \rho^k + 1 - \rho^k, \tag{3.25}$$

$$W_{m,n}(t) = \frac{\sum_{\{q,r\} \in S_{m,n}} W_{q,r}(t)}{|S_{m,n}|}, \tag{3.26}$$

where $S_{m,n}$ indicates the set of known parameters adjacent to the missing parameter being interpolated, ρ (with $0 \le \rho \le 1$) determines the trade-off between the interpolated value and the bias value as a function of iteration number k. After the first iteration, Figure 3.4b results. Repeating this process assigns estimates for $\hat{\alpha}_{m,n}(t)$ to all missing parameters (Figure 3.4c,d).

Next, a postprocessing step smooths all the matrix entries with a 5×5 gaussian kernel. Figure 3.5(a,b) shows respectively, an original set of known and missing parameters and the interpolated, smoothed parameters.

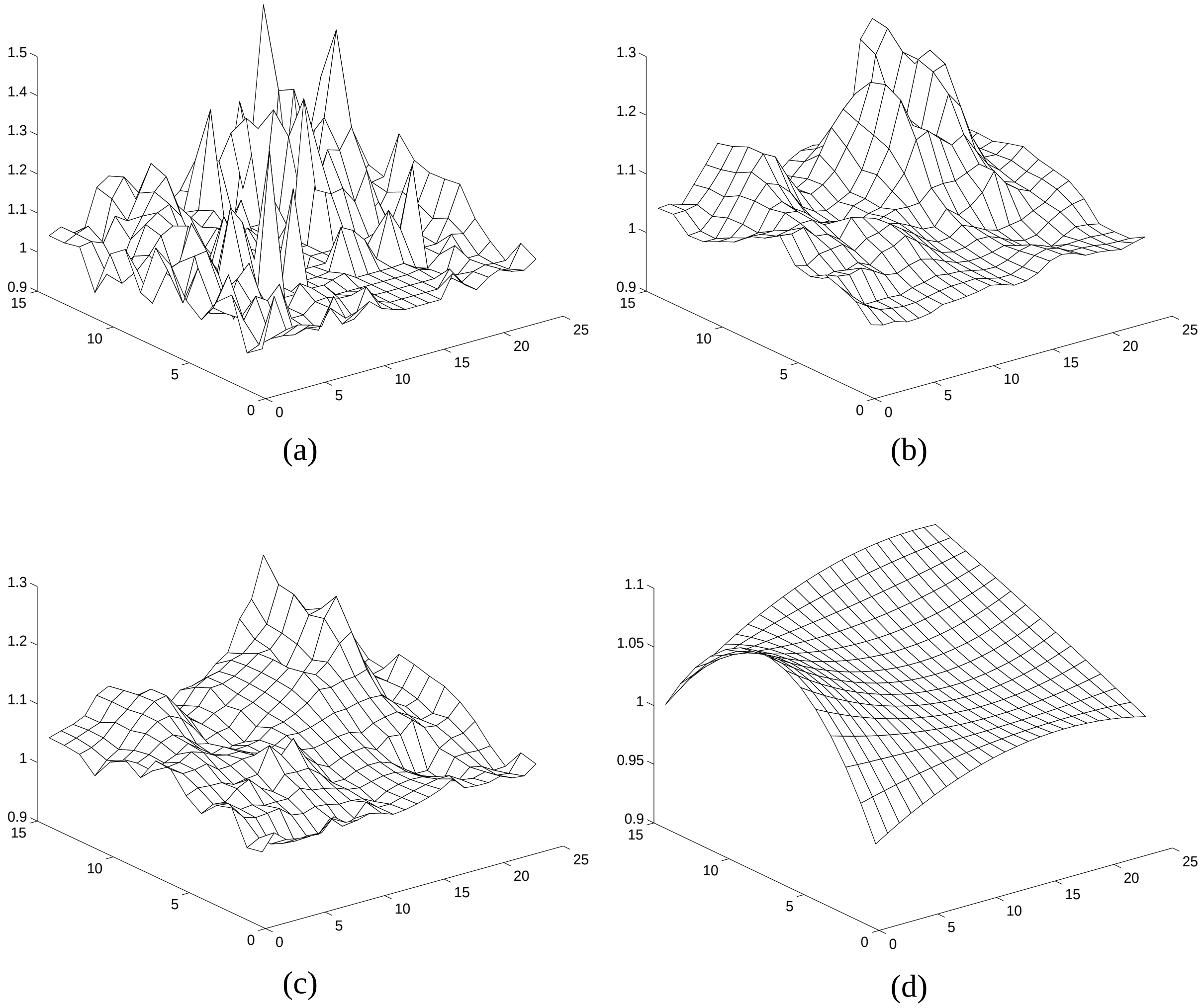
Interpolation by successive overrelaxation (SOR). SOR is a well-known iterative method based on repeated low-pass filtering [75]. Unlike the dilation technique, this method interpolates the missing parameters and smooths the known parameters simultaneously. SOR starts out with an initial approximation $\alpha^0_{m,n}(t)$. At each iteration k, the new solution $\alpha^{k+1}_{m,n}(t)$ is computed for all (m, n) by computing a residual term $r^{k+1}_{m,n}$ and subtracting this from the current solution:

$$\begin{aligned} r^{k+1}_{m,n,t} = {} & W_{m,n}(t) \cdot (\alpha^k_{m,n}(t) - \alpha^0_{m,n}(t)) + \\ & \lambda \cdot (4\alpha^k_{m,n}(t) - \alpha^k_{m-1,n}(t) - \alpha^k_{m+1,n}(t) - \alpha^k_{m,n-1}(t) - \alpha^k_{m,n+1}(t)), \end{aligned} \tag{3.27}$$

$$\alpha^{k+1}_{m,n}(t) = \alpha^k_{m,n}(t) - \varpi \cdot \frac{r^{k+1}_{m,n,t}}{W_{m,n}(t) + 4\lambda}. \tag{3.28}$$

Here $W_{m,n}(t)$ are the weights, λ determines the smoothness of the solution, and ϖ is the so-called overrelaxation parameter that determines the rate of convergence. The $\alpha^0_{m,n}(t)$ are initialized to the known multiplicative intensity-flicker parameters at (m, n), and to the bias value for the missing parameters.

The first term in (3.27) weighs the difference between the current solution and the original estimate, and the second term measures the smoothness. The solution is updated in (3.28) so that where the weights $W_{m,n}(t)$ are great, the original estimates $\alpha^0_{m,n}(t)$ are emphasized. In contrast, when the measurements are deemed less reliable, i.e., when $\lambda \gg W_{m,n}(t)$, emphasis is laid on achieving a smooth solution. This allows the generation of complete parameter fields where the known parameters, depending on their accuracy, are weighted and smoothed. Figure 3.5c shows results of this method.



(a) (b)

(c) (d)

Figure 3.5 (a) Set of original measurements with variable accuracy; the missing measurements have been set to 1, (b) parameters interpolated and smoothed by repeated dilation, (c) parameters interpolated and smoothed by SOR (250 iterations), (d) parameters interpolated and smoothed by polynomial fitting ($D_r = D_c = 2$). Note the differences in scale.

Interpolation by 2D polynomial fitting. By fitting a 2D polynomial $P(m, n, t)$ to the known parameters, the missing parameters can be interpolated and the known parameters are smoothed simultaneously. The 2D polynomial is given by [34]:

$$P(m, n, t) = \sum_{k=0}^{D_c} \sum_{l=0}^{D_r} c_{k,l,t} m^k n^l, \tag{3.29}$$

where D_c and D_r determine the degree of the polynomial surface and the coefficients $c_{k,l,t}$ shape the function. Polynomial fitting entails finding the coefficients $c_{k,l,t}$ so that the weighted mean squared difference of $P(m, n, t)$ and $\hat{\alpha}_{m,n}(t)$ is minimized for a given t:

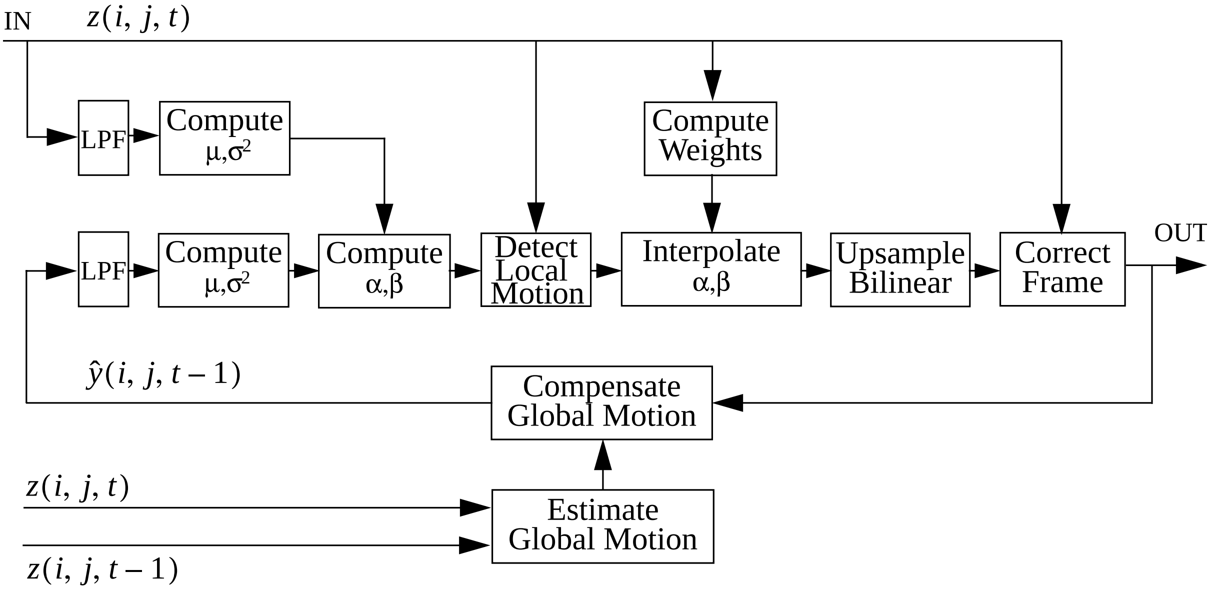


Figure 3.6 Global structure of the intensity-flicker correction system.

$$\min_{c_{k,l,t}} \left(\sum_{m,n} W_{m,n}(t) \cdot (P(m,n,t) - \hat{\alpha}_{m,n}(t))^2 \right). \tag{3.30}$$

The complexity of solving (3.30) is typical of a weighted least squares problem that requires computation of a pseudo inverse of a square matrix [92]. The number of columns (and rows) depends on the order of the polynomial and is determined by the number of coefficients $c_{k,l,t}$ at an instant t.

Biases are applied by setting the missing parameters to their bias value; the weights corresponding to these parameters are set to a fraction (e.g., one tenth) of the largest weight found for the known parameters. This will have little effect on the shape of the polynomial surface if only a few parameters are missing locally. Where many parameters are missing, the combined influence of the biased parameters will shape the polynomial locally towards the bias value.

The range of the results obtained by the dilation and SOR interpolation methods is limited to the range of the data. This is not the case for 2D polynomial fitting. The higher-order terms cause spurious oscillations if the order of the polynomial is taken too high, which leads to incorrect values for the interpolated and smoothed parameters. In practice, taking $D_c = D_r = 2$ gives the best results. Figure 3.5d shows a result of this interpolation and smoothing method.

3.4 Practical issues

Figure 3.6 shows the overall structure of the system of intensity-flicker correction. Some operations have been added in this figure that have not yet been mentioned. These operations

Image Blocks	Motion Detection	2D Polynomial	Successive OverRelaxation	Miscellaneous
Size: 30×20 Overlap: 1 pixel	$T_d = 5$ $D_{max} = 5$	$D_r = D_c = 2$	$\varpi = 1$ $\lambda = 5$	$\kappa = 0.85$ $var[\eta(x, y, t)] = 5$ $T_n = 25$

Table 3.1 Parameter settings of intensity-flicker correction system for the experiments.

improve the system's behavior. First, the current input and the previous system output (with global motion compensation) are low-pass filtered with a 5×5 gaussian kernel. Prefiltering suppresses the influence of high-frequency noise and the effects of small motion. Then, local means μ and variances σ^2 are computed to be used for estimating the intensity-flicker parameters. The estimated model parameters and the current input are used to detect local motions. Next, the missing parameters are interpolated and the known parameters are smoothed. Bilinear interpolation is used for upsampling the estimated parameters to full spatial resolution. The latter avoids the introduction of blocking artifacts in the correction stage that follows.

As mentioned in Section 3.2.3, the fact that a recursive structure is used for the overall system of intensity-flicker correction introduces the possibility of error propagation. Errors certainly do occur, for example, as a result of the need to approximate the expectation operator and from model mismatches. Therefore, it is useful to bias corrected intensities towards the contents of the current frame to avoid possible drift due to error accumulation. For this purpose, (3.2) is replaced by:

$$\hat{y}(\boldsymbol{i}) = \kappa \cdot (a(\boldsymbol{i}) \cdot z(\boldsymbol{i}) + b(i)) + (1 - \kappa) \cdot z(\boldsymbol{i}), \tag{3.31}$$

where κ is the forgetting factor. If $\kappa = 1$, the system relies completely on the frame corrected previously, and it tries to achieve the maximal reduction in intensity flicker. If $\kappa = 0$, we find that the system is switched off. A practical value for κ is 0.85.

3.5 Experiments and results

This section applies the system of intensity-flicker correction both to sequences containing artificially added intensity flicker and to sequences with real (non-synthetic) intensity flicker. This first set of experiments takes place in a controlled environment and evaluates the performance of the correction system under extreme conditions. The second set of experiments verifies the practical effectiveness of the system and forms a verification of the underlying assumptions of the approach presented in this chapter. The same settings for the system of intensity-flicker correction were used for all experiments to demonstrate the robustness of the approach (see Table 3.1).

Some thought should be given to what criteria are to be used to determine the effectiveness of the proposed algorithm. If the algorithm functions well and the image contents does not

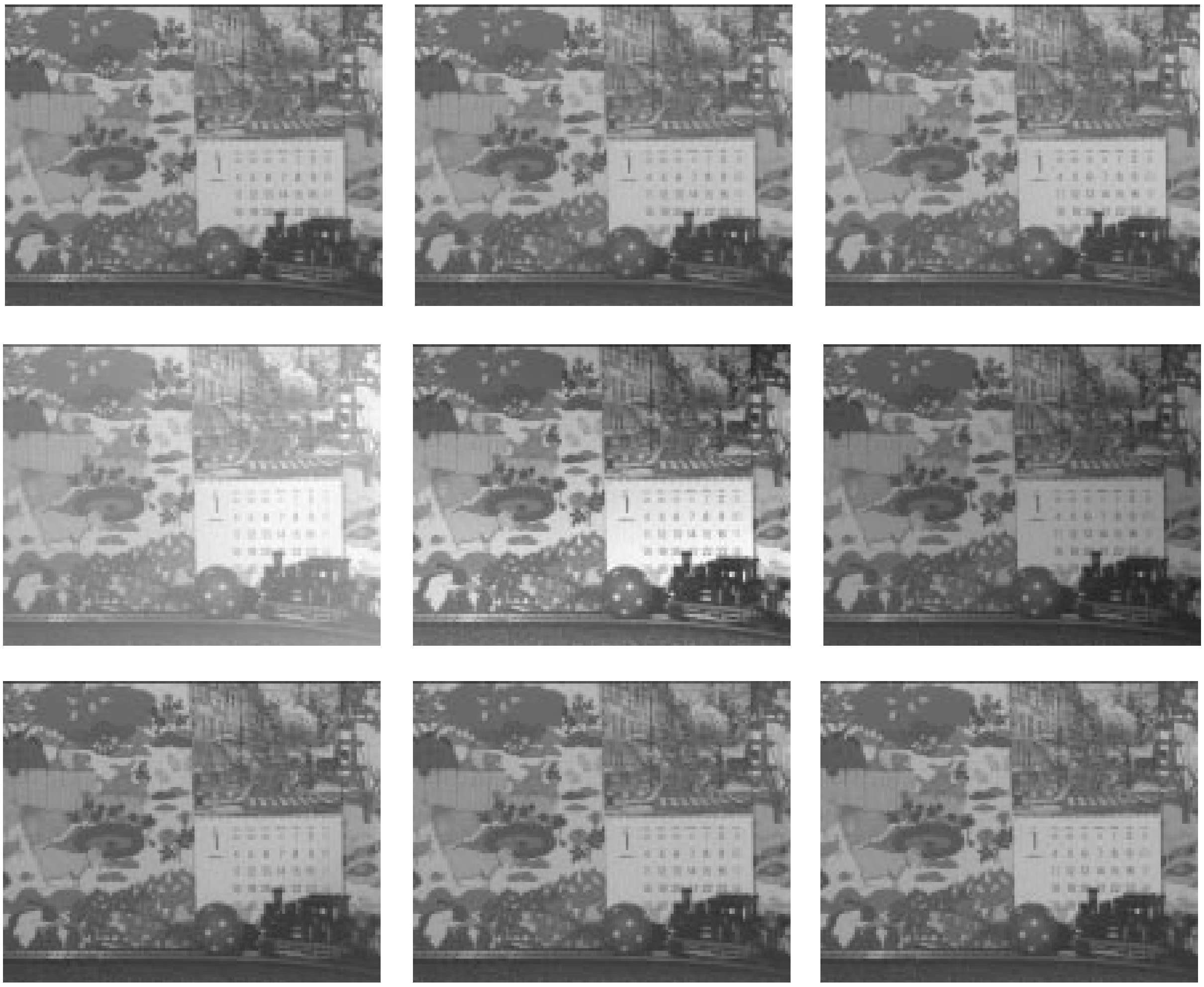


Figure 3.7 Top row: original frames 16, 17, and 18 of the *MobCal* sequence. Central row: degraded frames. Bottom row: frames corrected by the intensity-flicker correction system with successive overrelaxation.

change significantly, then the equalized frame means and variances should be similar from frame to frame. Indeed, the converse need not be true, but visual inspection helps to verify the results. Therefore, the temporal smoothness of frame means and frame variances measures the effectiveness of intensity-flicker correction system.

A sliding window approach is adopted here: the variance in frame mean and frame variance is computed locally over 24 frames (which corresponds to 1 second of film) and the estimated local variances are averaged over the whole sequence. There is a reason for using this sliding window. If the variation in frame means and variances are computed over long sequences, there are two components that determine the result: (1) variations due to flicker, and (2) variations due to changes in scene content. This thesis is only interested in the first component, which can be isolated by computing the variations over short segments.

3.5.1 Experiments on artificial intensity flicker

For the first set of experiments the *Mobile* sequence (40 frames), containing moving objects

	MobCal		Soldier		Mine		Charlie	
	Mean	Var.	Mean	Var.	Mean	Var.	Mean	Var.
Degraded	19.8	501	2.7	44	2.3	61	8.5	435
Dilation	5.5	110	0.8	29	1.0	37	5.6	319
SOR	5.2	86	0.8	31	1.0	40	4.9	235
2D Polynomial	5.8	105	0.9	27	1.2	41	6.3	333

Table 3.2 Standard deviation of averaged frame mean and frame variance of degraded and sequences corrected by various interpolators in the intensity flicker correction system.

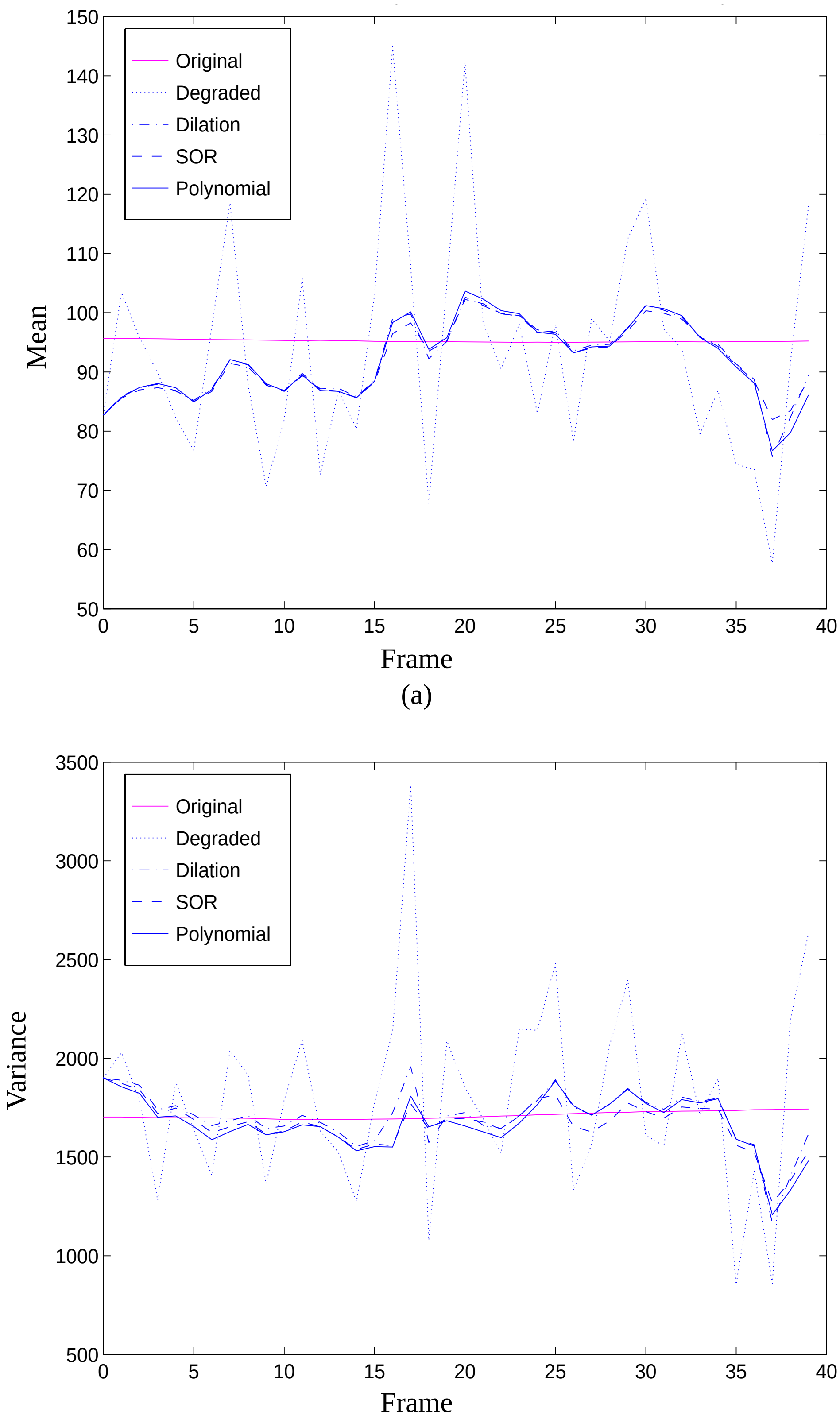
and camera panning (0.8 pixels/frame), is used. Artificial intensity flicker was added to this sequence according to (3.1). The intensity-flicker parameters were artificially created from 2D polynomials, defined by (3.29), with degree $D_r = D_c = 2$. The coefficients $c_{k, l, t}$ are drawn from the normal distribution $N(0, 0.1)$, and from $N(1, 0.1)$ for $c_{0, 0, t}$, to generate the $\alpha(\boldsymbol{i})$ and from $N(0, 10)$ to generate the $\beta(\boldsymbol{i})$. Visually speaking, this leads to a severe amount of intensity flicker (Figure 3.7).

The degraded sequence is corrected three times, and each time a different interpolation and smoothing algorithm is used, as described in Section 3.3.3. Figure 3.7 shows some corrected frames. Figure 3.8 plots the frame means and the frame variances of original, degraded and corrected sequences. It can be seen from these graphs that the variations in frame mean and variance have been strongly reduced. Visual inspection confirms that the amount of intensity flicker has been reduced significantly. However, residues of local intensity flicker are clearly visible when the dilation interpolation method is used. The SOR interpolation method gives the best visual results.

Table 3.2 lists the standard deviation of the frame means and frame variances computed over short segments by the sliding window approach and averaged as mentioned before. This table shows that the artificial intensity flicker severely degraded the sequence. It also shows that the intensity-flicker correction system strongly reduces fluctuations in frame mean and frame variance. The SOR interpolation method gives the best numerical results.

3.5.2 Experiments on naturally degraded film sequences

Three sequences from film archives were used for the second set of experiments. Table 3.2 lists the results. The first sequence, called *Soldier*, is 226 frames long. It shows a soldier entering the scene through a tunnel. There is some camera unsteadiness during the first 120 frames, then the camera pans to the right and up. There is film-grain noise and a considerable amount of intensity flicker in this sequence. The total noise variance was estimated to be 8.9 by the method described in [58]. Figure 3.9 shows three frames from this sequence, original and corrected. Figure 3.10 indicates that the fluctuation in frame means and variances have significantly been reduced by the intensity-flicker correction system. Visual inspection shows that all three methods significantly reduce the intensity flicker without introducing visible new artifacts. The best visual results are obtained with the SOR interpolation method.



(a)

(b)

Figure 3.8 (a) Frame means and (b) variances of original MobCal sequence, MobCal sequence with artificial intensity flicker, and sequences corrected by various interpolation and smoothing methods within the system for intensity-flicker correction.

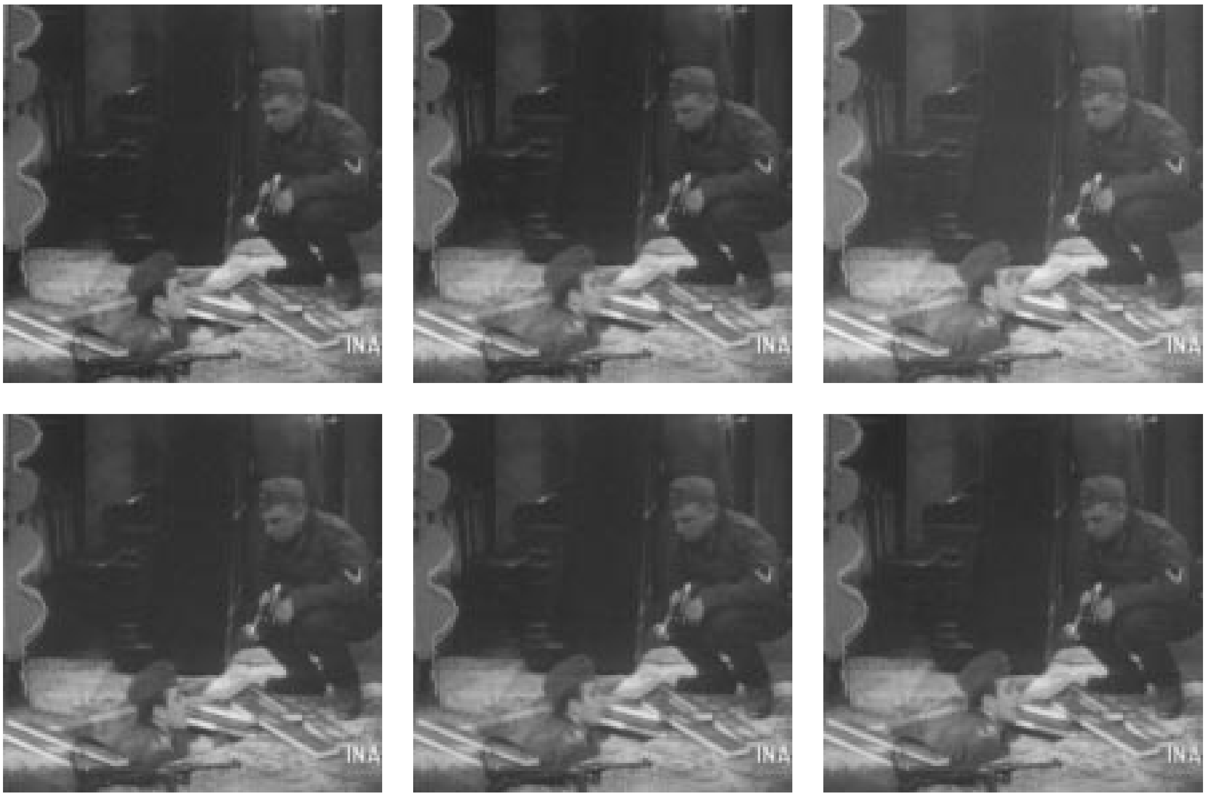
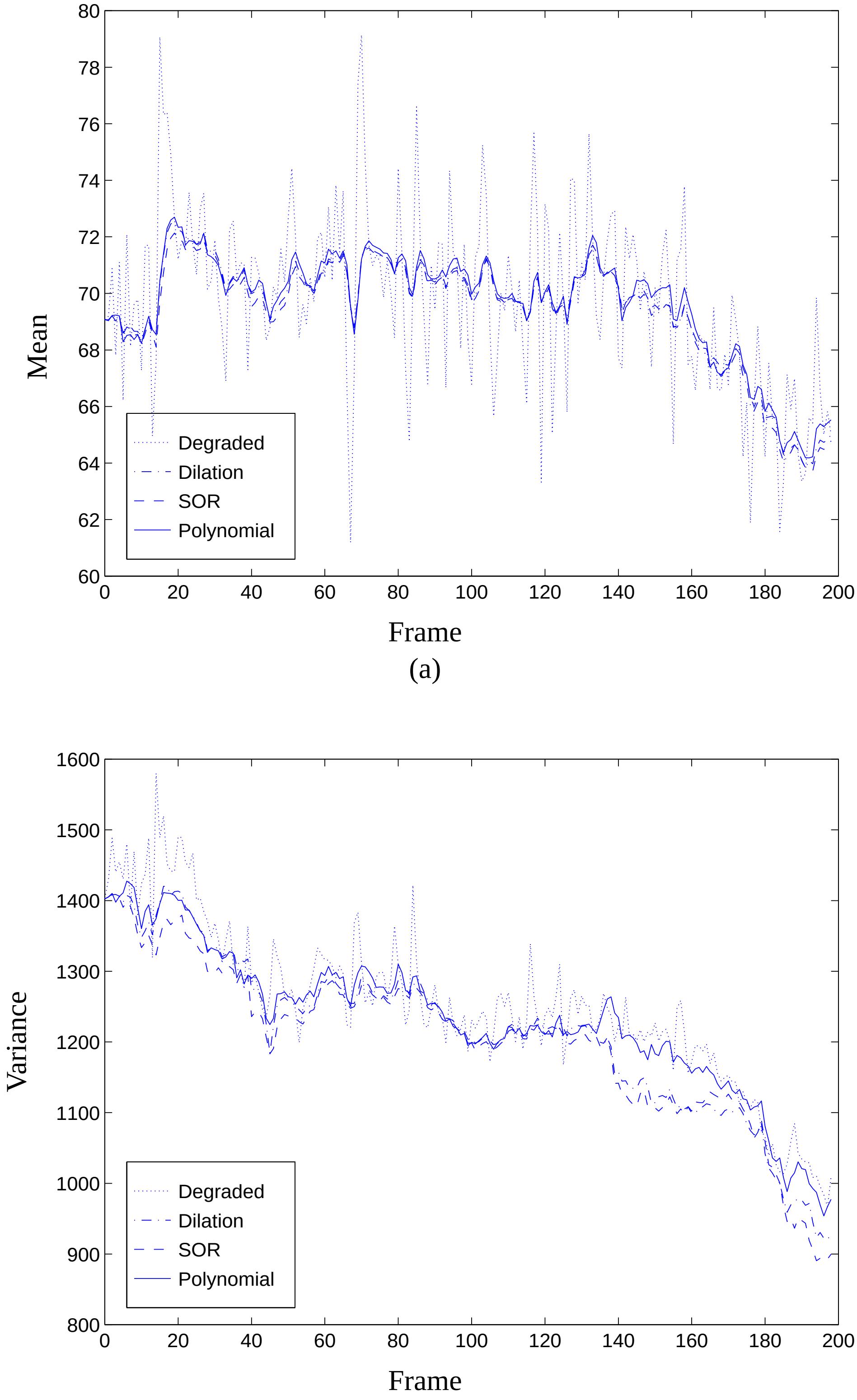


Figure 3.9 Top: frames 16, 17, and 18 of the naturally degraded Soldier sequence. Bottom: frames corrected by the intensity-flicker correction system using the 2D polynomial interpolation method.

The second naturally degraded sequence, called *Mine*, consists of 404 frames. This sequence depicts people in a mine. It contains camera pan, some zoom, and it is quite noisy (estimated noise variance 30.7**)**. The intensity flicker is not as severe as in the *Soldier* sequence. Figure 3.11 shows the frame means and variances of the degraded and the corrected sequences. Visually, the results obtained from the dilation interpolation method show some flickering patterns. The 2D polynomial interpolation leaves some flicker near the edges of the picture. The SOR method shows good results.

The third sequence is a clip of 48 frames from a Charlie Chaplin film, called *Charlie*. Some frames have so much intensity flicker that it looks as if the film has been overexposed and the texture is lost completely in some regions. Besides intensity flicker, this sequence is characterized by typical artifacts occurring in old films, such as blotches, scratches, and noise (estimated variance 5.0). Figure 3.12 shows that the fluctuations in frame means and variances have diminished. Again, from a subjective point of view, the SOR interpolation technique gives the best result, but a slight loss of contrast is noted in the corrected sequence.

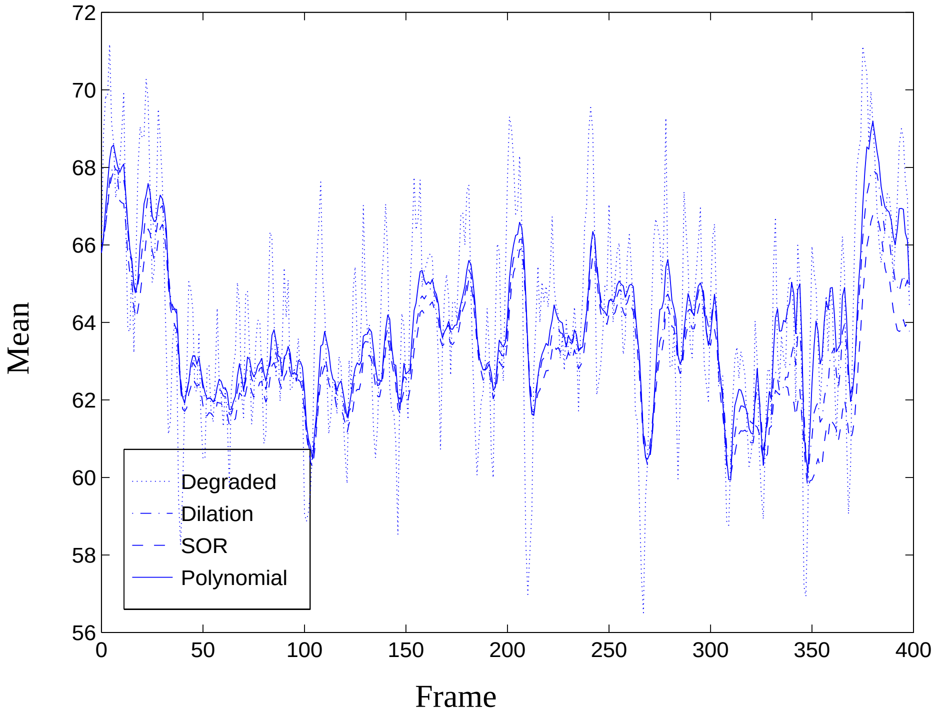
Table 3.2 indicates that the intensity-flicker correction system significantly reduces the fluctuations in frame mean and frame variance of all the test sequences. The SOR interpolation method gives the best numerical results: in all cases it gives the largest reduction in variation of the mean image intensity and it gives a reduction in variation of image variance that is similar or better than that obtained by the other interpolation methods.



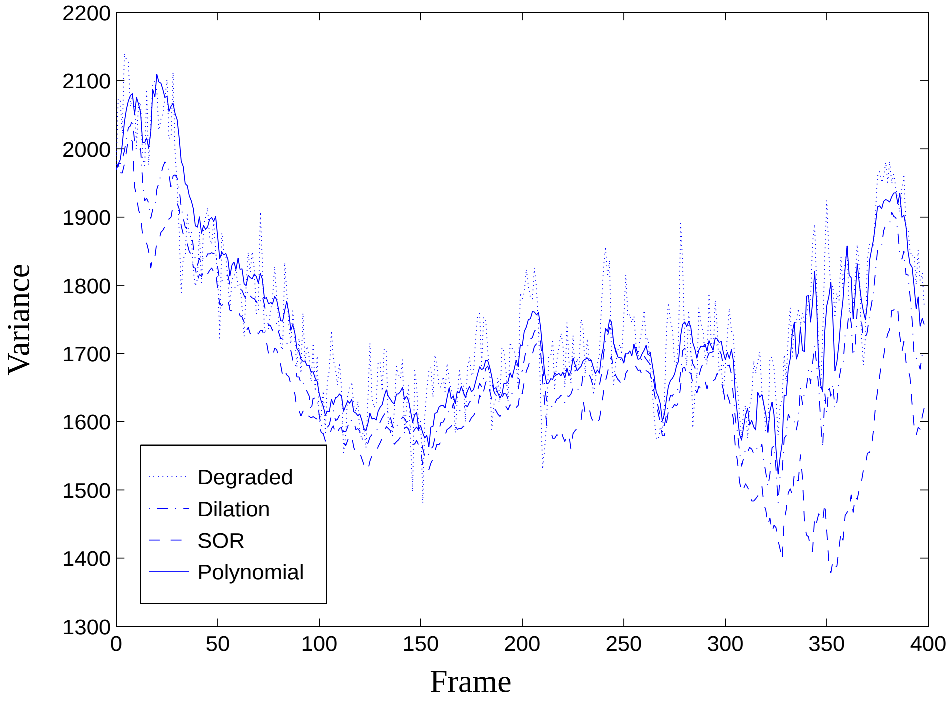
(a)

(b)

Figure 3.10 Frame means (a) and variances (b) of the naturally degraded Soldier sequence and sequences corrected by the system for intensity-flicker correction with various interpolation and smoothing methods.

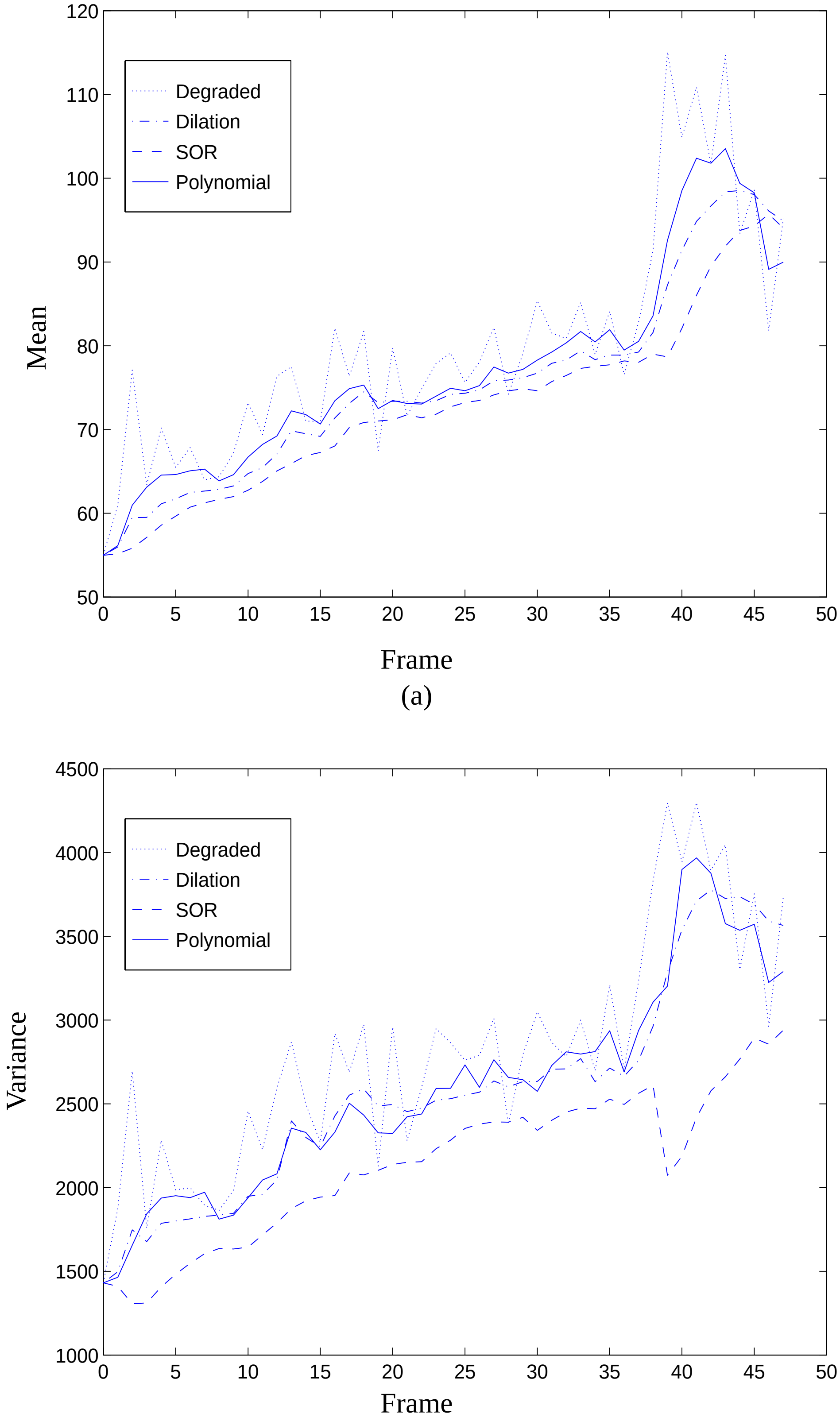


(a)



(b)

Figure 3.11 Frame means (a) and variances (b) of the naturally degraded Mine sequence and sequences corrected by the system for intensity-flicker correction with various interpolation and smoothing methods.



(a)

(b)

Figure 3.12 Frame means (a) and variances (b) of the naturally degraded Charlie sequence and sequences corrected by the system for intensity-flicker correction with various interpolation and smoothing methods.

3.6 Discussion

This chapter introduced a novel method for removing intensity flicker from image sequences that significantly reduces the temporal fluctuations in local image mean and variance. The system is based on simple block-based operations and motion detection. Therefore the complexity of the system is limited. This is advantageous for real-time implementation in hardware.

Improvements to the system are certainly conceivable. For instance, effort could be put into reducing the sizes of the image regions for which estimated flicker parameters are discarded due to local motion. In the current scheme, data in whole image blocks are discarded even though large parts of those blocks may not have been affected by motion. Alternatively, instead of detecting motion, an approach that incorporates robust motion estimation into the flicker correction system could be developed. This would result in a system for simultaneous motion and parameter estimation and intensity-flicker correction.

A hardware implementation based on the system described in this chapter was realized by the AURORA project. The system proved capable of restoring a large number of flickering image sequences successfully.

Chapter 4

Blotch detection and correction

Summary. Blotches are common artifacts in old film sequences. The first part of this chapter reviews existing methods for blotch detection and correction. The second part concentrates on developing improved techniques for blotch detection. Such techniques take into account the influence of noise on the detection process; they also exploit the spatial coherency inherent to blotches. The third part of this chapter presents a new, fast, model-based method for good-quality interpolation of blotched data. This method is faster than existing model-based interpolators. It is also more robust to corruption in the reference data that is used by the interpolation process. The performance of the resulting system for blotch detection and correction is evaluated with test sequences.

4.1 System for blotch detection and correction

Blotches are artifacts typically related to film. The loss of gelatin and dirt particles covering the film cause blotches. The original intensities corrupted by blotches are lost and will be referred to as *missing data*. Correcting blotches entails detecting the blotches and interpolating the missing data from data that surround the corrupted image region. The use of temporal information often improves the quality of the results produced by the interpolation process. This means that reference data from which the missing data are interpolated, need to be extracted from frames preceeding and/or following the frame currently being restored. Motion estimation and compensation is required to obtain optimal interpolation results.

The methods for blotch detection presented in this chapter assume the degradation model from (2.2), either implicitly or explicitly [47]:

$$z(\boldsymbol{i}) = (1 - d(\boldsymbol{i})) \cdot y(\boldsymbol{i}) + d(\boldsymbol{i}) \cdot c(\boldsymbol{i}), \tag{4.1}$$

where $z(\boldsymbol{i})$ and $y(\boldsymbol{i})$ are the observed and the original (unimpaired) data, respectively. The binary blotch detection mask $d(\boldsymbol{i})$ indicates whether each pixel has been corrupted: $d(\boldsymbol{i}) \in \{0, 1\}$. The values at the corrupted sites are given by $c(\boldsymbol{i})$, with $c(\boldsymbol{i}) \neq y(\boldsymbol{i})$. One property of blotches is the smooth variation in intensity values at the corrupted sites; the variance $c(\boldsymbol{i})$ within a blotch is small. Blotches seldom appear at the same location in a pair of consecutive frames. Therefore the binary mask $d(\boldsymbol{i})$ will seldom be set to one at two spatially co-sited locations for a pair of consecutive frames. However, there is spatial coherence within a blotch; if a pixel is blotched, it is likely that some of its neighbors are corrupted as well, i.e., if $d(\boldsymbol{i}) = 1$ it likely that some other $d(i \pm 1, j \pm 1, t) = 1$ also.

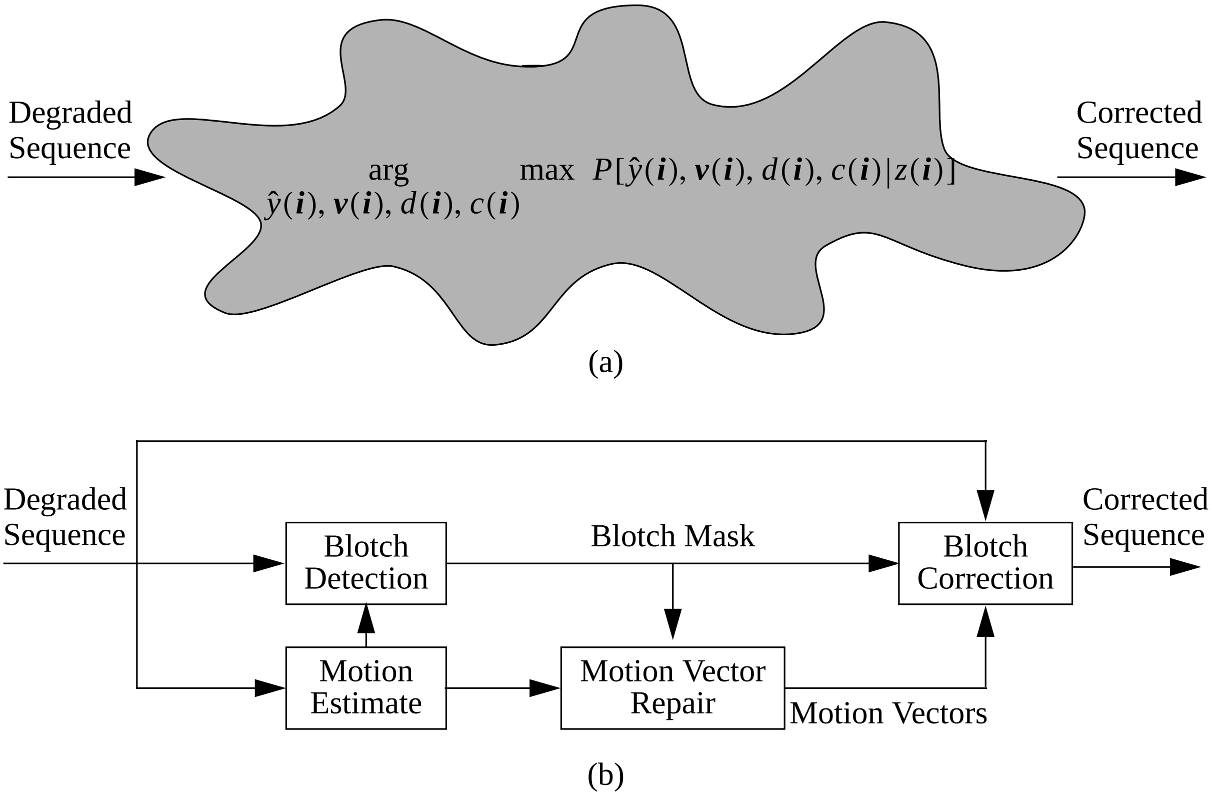
The following sections use various models for the original, uncorrupted image data. The common element is that these models do not allow large temporal discontinuities in image intensity along the motion trajectories. This constraint results from the fact that $c(\boldsymbol{i}) \neq y(\boldsymbol{i})$ in the degradation models, which implies that blotches introduce temporal discontinuities in image intensity. Temporal discontinuities in image intensity are also caused by moving objects that cover and uncover the background. There is a difference between the effects of blotches and the effects of motions. Motion tends to cause temporal discontinuities in either the forward or the backward temporal direction, but not in both directions at the same time. Blotches cause discontinuities simultaneously in both temporal directions.

The estimated motion vectors are unreliable at image locations corrupted by blotches because they are determined with incorrect, corrupted data. Models for motion vector repair and for blotch correction assume a relationship between the original image data at the corrupted sites and the data surrounding those sites (temporally and/or spatially). For example, for motion vector repair, this relationship can be smoothness of the motion vector field. For blotch correction, this relationship can be defined by *autoregressive* (AR) image models.

Figure 4.1 illustrates two possible approaches for detecting and correcting blotches. The first approach computes the locations of the blotches, the motion vectors, and the corrected intensities simultaneously within a single bayesian framework. *Maximum a posteriori* (MAP) estimates for the true image intensities, $\hat{y}(z)$, the motion vectors $\boldsymbol{v}(\boldsymbol{i})$, the blotch detection mask $d(\boldsymbol{i})$ and the intensities of the blotches $c(\boldsymbol{i})$ are computed from the observed images $z(\boldsymbol{i})$:

$$\underset{\hat{y}(\boldsymbol{i}),\, \boldsymbol{v}(\boldsymbol{i}),\, d(\boldsymbol{i}),\, c(\boldsymbol{i})}{\arg\max}\; P[\hat{y}(\boldsymbol{i}), \boldsymbol{v}(\boldsymbol{i}), d(\boldsymbol{i}), c(\boldsymbol{i}) | z(\boldsymbol{i})]. \tag{4.2}$$

This is an elegant framework because it defines an optimal solution that takes dependencies between the various parameters into account. It was applied successfully in [47]. A disadvantage of this method, besides its great computational complexity, is the difficulty of determining what influence the individual assumptions for the likelihood functions and priors have on the final outcome of the overall system. Hence, it is difficult to determine whether the assumed priors and likelihood functions give optimal results.


Degraded Sequence
arg max $P[\hat{y}(\boldsymbol{i}), \boldsymbol{v}(\boldsymbol{i}), d(\boldsymbol{i}), c(\boldsymbol{i}) | z(\boldsymbol{i})]$
$\hat{y}(\boldsymbol{i}), \boldsymbol{v}(\boldsymbol{i}), d(\boldsymbol{i}), c(\boldsymbol{i})$
Corrected Sequence
Degraded Sequence
Blotch Detection
Blotch Mask
Blotch Correction
Corrected Sequence
Motion Estimate
Motion Vector Repair
Motion Vectors

(a)

(b)

Figure 4.1 (a) Simultaneous approach for blotch detection and correction vs. (b) modular approach.

The second approach towards resolving blotches is a modular approach, as shown in Figure 4.1b. The *motion estimate* module estimates motion between consecutive frames in the forward and backward directions (from t to $t+1$ and from t to $t-1$ respectively). On the basis of motion estimates and the incoming degraded data, the *blotch detection* module detects blotches. The *motion vector repair* module corrects faulty motion vectors. Finally, the *blotch correction* module corrects blotches using the corrected motion vectors, the binary blotch detection mask, and the degraded image sequence.

This chapter concentrates on the modular approach for blotch detection and correction. This approach has the advantage that the modules can be designed and evaluated independently of each other. Furthermore, the modular approach has the advantage of being computationally much less demanding than the simultaneous bayesian approach.

This chapter is structured as follows. Section 4.2 reviews existing techniques for blotch detection, motion vector repair and blotch correction. Section 4.3 introduces a new technique for improving the detection results by postprocessing blotch detection masks. The postprocessing operations significantly reduce the number of false alarms that are inherent to any detection problem. Section 4.4 shows that increasing the temporal aperture of a detector gives significant gains in some cases. Section 4.5 presents a new, fast model-based method for excellent quality of missing data interpolation. Section 4.6 evaluates the performance of the complete blotch removal system and concludes this chapter.

4.2 Overview of existing techniques

4.2.1 Blotch detection techniques

The parameter estimation problem for the degradation model consists of determining the binary blotch detection mask $d(\boldsymbol{i})$ for each frame. If required, $c(\boldsymbol{i})$ can easily be found once $d(\boldsymbol{i})$ is known. The blotch detectors presented in this section all apply the same principle: they check whether the observed data $z(\boldsymbol{i})$ fit an image model for $y(\boldsymbol{i})$. If this is not the case, the image is assumed to be corrupted and a blotch is flagged.

<u>SDIa detector.</u> The *spike detection index-a* (SDIa) is a simple heuristic method for detecting temporal discontinuities in image intensity [47,48]. It compares each pixel intensity of the current frame $z(\boldsymbol{i})$ to the corresponding intensities in the forward and backward temporal directions by computing the minimum squared difference $SDIa(\boldsymbol{i})$:

$$SDIa(\boldsymbol{i}) = min[(z(\boldsymbol{i}) - z_{mc}(\boldsymbol{i}, t+1))^2, (z(\boldsymbol{i}) - z_{mc}(\boldsymbol{i}, t-1))^2]. \tag{4.3}$$

Large values for $SDIa(\boldsymbol{i})$ indicate discontinuities in image intensity in both the forward and backward temporal directions. A blotch is detected if $SDIa(\boldsymbol{i})$ exceeds a threshold T_1:

$$d_{SDIa}(\boldsymbol{i}) = \begin{cases} 1 & \text{if} \quad SDIa(\boldsymbol{i}) > T_1 \\ 0 & \text{otherwise} \end{cases} \qquad \text{with } T_1 \geq 0, \tag{4.4}$$

where T_1 is a threshold selected by the user. If a small value is chosen for this threshold, the detector is very sensitive and will detect a large percentage of the blotches corrupting an image. However, due to the great sensitivity, many false alarms will result as well. Increasing the value of T_1 reduces the sensitivity; it reduces both the number of false alarms and the number of correct detections.

A variation on the SDIa detector is the SDIp detector. SDIp has an additional constraint that requires the signs of $z(\boldsymbol{i}) - z_{mc}(\boldsymbol{i}, t+1)$ and $z(\boldsymbol{i}) - z_{mc}(\boldsymbol{i}, t{-}1)$ to be identical before a blotch can be detected. This constraint reduces the number of false alarms resulting from erroneous motion estimates. In the case of correct motion estimation, the reference pixels in the previous and next frames are assumed to be identical, and therefore the intensity differences with the corrupted data in the current frame should have the same polarity. Note that this is not necessarily true in the case of occlusion and noisy data.

<u>ROD detector.</u> The *rank-ordered differences* (ROD) detector is a heuristic detector based on *order statistics* (OS) [64]. Let p_k with $k = 1, 2, \ldots, 6$ be a set of reference pixels relative to a pixel from $z(\boldsymbol{i})$. These reference pixels are taken from the motion compensated previous and next frames at locations spatially co-sited with pixel $z(\boldsymbol{i})$ and its two closest vertical neighbors (see Figure 4.2a). Let r_m be the reference pixels p_k ordered by rank with $r_1 \leq r_2 \leq \ldots \leq r_6$. The rank order mean r_{mean} and rank-order differences $ROD(\boldsymbol{i}, l)$ with $l = 1, 2, 3$ are defined by (see Figure 4.2b):

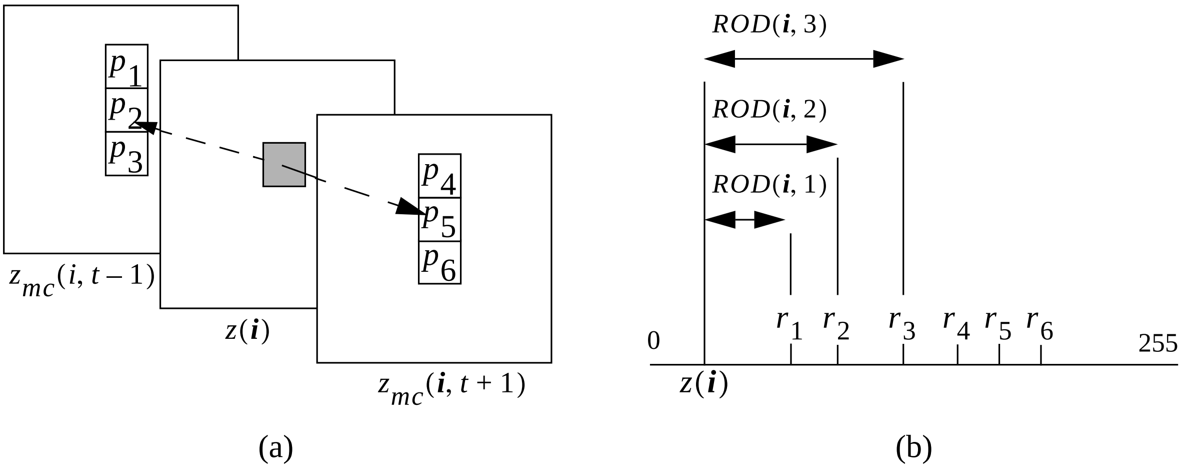


Figure 4.2 (a) Selection of reference pixels p_k from motion compensated previous and next frames, (b) computation of $ROD(\boldsymbol{i}, l)$ based on pixels p_k ordered by rank: r_m.

$$r_{mean} = \frac{r_3 + r_4}{2}, \tag{4.5}$$

$$ROD(\boldsymbol{i}, l) = \begin{cases} r_l - z(\boldsymbol{i}) & \text{if } z(\boldsymbol{i}) \le r_{mean} \\ z(\boldsymbol{i}) - r_{7-l} & \text{if } z(\boldsymbol{i}) > r_{mean} \end{cases} \quad \text{with } l = 1, 2, 3. \tag{4.6}$$

A blotch is detected if at least one of the rank-order differences exceeds a specific threshold T_l. The T_l are set by the user and determine the detector's sensitivity:

$$d_{ROD}(\boldsymbol{i}) = \begin{cases} 1 & \text{if } ROD(\boldsymbol{i}, l) > T_l \\ 0 & \text{else} \end{cases} \quad \text{with } 0 \le T_1 \le T_2 \le T_3 \text{ and } l = 1, 2, 3. \tag{4.7}$$

MRF detector. In [48] an a posteriori probability for a binary occlusion map, given the current frame and a motion-compensated reference frame, is defined. The occlusion map indicates whether objects in the current frame are also visible in a reference frame. The *probability mass function* (pmf) for the a posteriori probability of the occlusion map is given by:

$$P[d_k(\boldsymbol{i}) | z(\boldsymbol{i}), z_{mc}(\boldsymbol{i}, t+k)] \propto P[z(\boldsymbol{i}) | d_k(\boldsymbol{i}), z_{mc}(\boldsymbol{i}, t+k)] \cdot P[d_k(\boldsymbol{i})], \tag{4.8}$$

where the symbol $\propto$ means *is proportional to,* and k indicates which reference frame is used. Maximizing (4.8) gives the MAP estimate for an occlusion mask.

Blotches are detected where occlusions are detected both in forward and backward temporal directions; $k = 1$ and $k = -1$:

$$d_{MRF}(\boldsymbol{i}) = \begin{cases} 1 & \text{if } (d_1(\boldsymbol{i}) = 1) \wedge (d_{-1}(\boldsymbol{i}) = 1) \\ 0 & \text{otherwise} \end{cases}. \tag{4.9}$$

The likelihood function in (4.8) is defined by:

$$P[z(\boldsymbol{i})|d_k(\boldsymbol{i}), z_{mc}(\boldsymbol{i}, t+k)] \propto \exp\left(-\sum_{\boldsymbol{i} \in S} [(1-d_k(\boldsymbol{i})) \cdot (z(\boldsymbol{i}) - z_{mc}(\boldsymbol{i}, t+k))^2]\right), \tag{4.10}$$

where S indicates the set of all spatial locations within a frame. This likelihood function indicates that, in the absence of occlusion, $d_k(\boldsymbol{i}) = 0$, the squared difference between the current pixel $\boldsymbol{i}$ and the corresponding pixel from the motion-compensated reference frame is likely to be small. The prior in (4.8) is given by:

$$P[d_k(\boldsymbol{i})] \propto \exp\left(-\sum_{\boldsymbol{i} \in S} [\beta_1 \cdot f(d_k(\boldsymbol{i})) + \beta_2 \cdot d_k(\boldsymbol{r})]\right) \qquad \text{with } \beta_1, \beta_2 \geq 0, \tag{4.11}$$

where the function $f(d_k(\boldsymbol{i}))$ counts the number of neighbors of $d_k(\boldsymbol{i})$ that are different from $d_k(\boldsymbol{i})$. The term $\beta_1 \cdot f(d_k(\boldsymbol{i}))$ in (4.11) constrains the occlusion map to be consistent locally. If an occlusion mask is locally inconsistent, $\beta_1 \cdot f(d_k(\boldsymbol{i}))$ is large and the probability of $P[d_k(\boldsymbol{i})]$ is made smaller. The term $\beta_2 \cdot d_k(\boldsymbol{r})$ in (4.11) is a penalty term that suggests that it is unlikely that many pixels are occluded. The user controls the strength of the self-organization and the sensitivity of the detector by selecting values for β_1 and β_2.

Combining (4.8), (4.10), and (4.11) gives:

$$\begin{gathered} P[d_k(\boldsymbol{i})|z(\boldsymbol{i}), z_{mc}(\boldsymbol{i}, t+k)] \propto \\ \exp\left(-\sum_{\boldsymbol{i} \in S} [(1-d_k(\boldsymbol{i})) \cdot (z(\boldsymbol{i}) - z_{mc}(\boldsymbol{i}, t+k))^2 + \beta_1 \cdot f(d_k(\boldsymbol{i})) + \beta_2 \cdot d_k(\boldsymbol{i})]\right). \end{gathered} \tag{4.12}$$

Equation (4.12) can be maximized with *simulated annealing* (SA) [28]. It is maximized once for $k = 1$ and once for $k = -1$. The resulting occlusion masks are combined by (4.9) to give the binary blotch detection mask $d_{MRF}(\boldsymbol{i})$.

AR detector. The assumptions that underlie the AR detector are that uncorrupted images follow AR models and that the images can be predicted well from the motion compensated preceeding and/or following frames [48]. If the motion-compensated frame at $t + k$ is used as a reference, the observed current frame $z(\boldsymbol{i})$ is given by:

$$\begin{aligned} z(\boldsymbol{i}) &= \sum_{l=1}^{n} a_l \cdot z_{mc}(\boldsymbol{i} + \boldsymbol{q}_l, t + k) + e(\boldsymbol{i}, t + k) \\ &= \hat{z}_k(\boldsymbol{i}) + e(\boldsymbol{i}, t + k), \end{aligned} \tag{4.13}$$

where the a_l are the n AR model coefficients estimated from the observed data (see, for example, [94]), $\boldsymbol{q}_k$ give the relative positions of the reference pixels with respect to the current pixel and $e(\boldsymbol{i}, t + k)$ denotes the prediction error.

In the absence of blotches and occlusion, the prediction errors $e(\boldsymbol{i}, t + k)$ are small. A blotch is detected if the squared prediction error exceeds a user defined threshold T_1 in both the forward ($k = 1$) and backward ($k = -1$) directions:

$$d_{AR}(\boldsymbol{i}) = \begin{cases} 1 & \text{if } (e^2(\boldsymbol{i}, t+1) > T_1) \ \wedge \ (e^2(\boldsymbol{i}, t-1) > T_1) \\ 0 & \text{otherwise} \end{cases} \quad \text{with } T_1 \geq 0. \tag{4.14}$$

Evaluation. To compare the effectiveness of the detectors described in this section, Figure 4.3 plots their *receiver operator characteristics* (ROCs) for four test sequences. An ROC plots the false alarm rate versus the correct detection rate of a detector. Ideally, the ratio of correct detections to false alarms is large. For the SDIa, ROD, and AR detectors, the curves were obtained by letting T_1 vary so that $1 \leq T_1 \leq 35$ (for the ROD detector, $T_2 = 39$ and $T_3=55$ were used). For the AR detector, the image was subdivided into blocks of 28×28 pixels, and a set of AR coefficients was computed for each block. The support consisted of five pixels as in [48] (see Figure 4.4). For the MRF detector, $3 \leq \beta_1 \leq 8$ and $9 \leq \beta_2 \leq 1369$ were used.

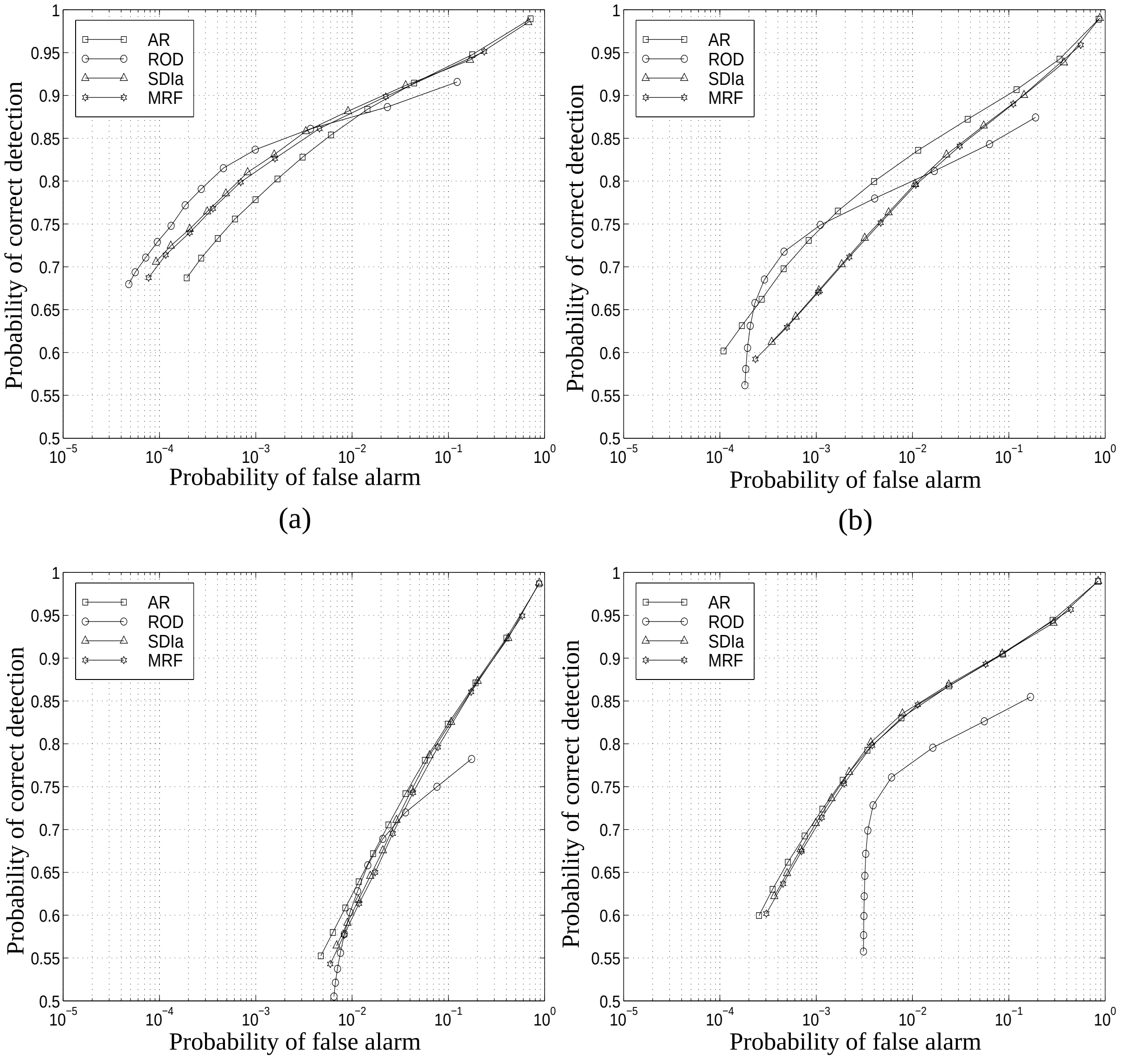
The detectors were applied to four test sequences, namely *Western*, which was also used in [48], *Mobcal*, *Manege*, and *Tunnel*. To avoid problems caused by the combination of interlacing and fast motion, only the odd fields from the last two sequences were used[1].

All sequences were degraded by adding artificial blotches. Each artificial blotch had a fixed gray value that was drawn uniformly between 16 and 240, which is the allowed range for pixel intensities in this thesis. The *Western* sequence originates from film and therefore contains granular noise. The *MobCal*, *Manege*, and *Tunnel* sequences, which were recorded by modern cameras, have little noise. To let them resemble real film data more closely, white gaussian noise with variance 10 was added after the blotches were added. Therefore, for these sequences, unlike for the *Western* sequence, the blotches are no longer completely smooth. Motion was estimated by an hierarchical motion estimator (Appendix A).

Figure 4.3 shows that the performance of the detectors strongly depends on the sensitivity to which they are set and on the sequences themselves. The best detection results are obtained for the *Western* sequence, which has relatively low local contrasts. The poorest results are obtained for the *Manege*, sequence which contains fast motion and sharp local contrasts.

---

[1] This is reasonable because blotches are artifacts that are typically related to film with no interlacing.



(a) (b)

(c) (d)

Figure 4.3 Receiver operator characteristics for various blotch detectors for (a) Western sequence, (b) MobCal sequence, (c) Manege sequence, (d) Tunnel sequence.

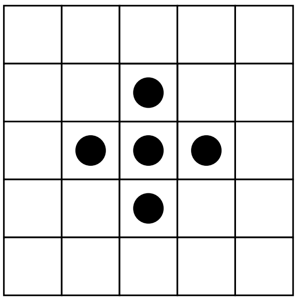


Figure 4.4 Support (circles) from reference frame at $t+k$ used for AR detector. The center of the support is aligned with pixel being processed in the current frame t.

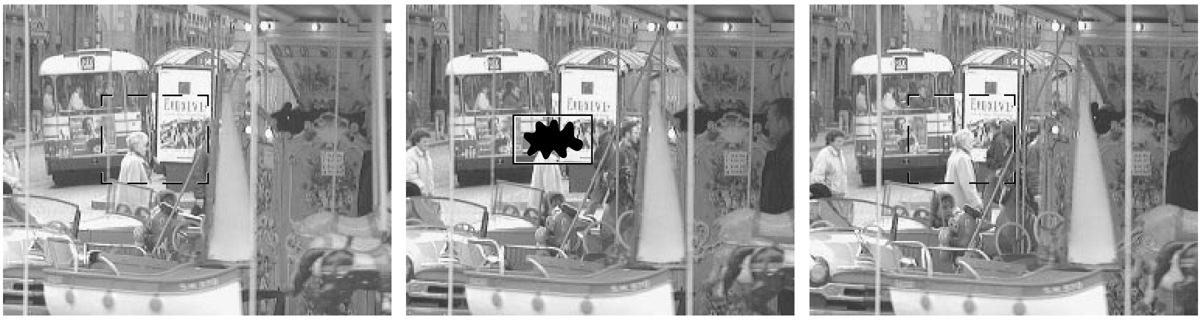


Figure 4.5 Three frames from the Manege sequence with a single blotch (black) in the central frame and a bounding box. The regions within the dashed boxes in the outermost frames indicate the search region for the block matcher.

The experiments show that no detector consistently outperforms any other. In some instances, the AR detector shows the best performance; in other instances, it shows the poorest performance. It can been seen from the ROCs that greater complexity does not necessarily lead to better results. The SDIa detector requires only a fraction of the number of computations required by the MRF detector, and both give a similar performance for all sequences. The ROD detector performs well for most sequences. However, it breaks down in the *Tunnel* sequence. This is because many false alarms are generated in this sequence as a result of the fixed settings chosen for T_2 and T_3.

4.2.2 Techniques for motion vector repair

Estimated motion vectors are less reliable when an image is blotched. Hence, the reference data extracted from the motion-compensated reference frames and used for interpolating the missing data may be erroneous. Motion vector repair can improve the likelihood of obtaining correct reference data. Motion vector repair has been investigated in the context of error concealment in (compressed) digital video transmission where each 8×8 or 16×16 image block has one motion vector assigned to it.

Two basic approaches to motion vector repair are found in literature. The first approach re-estimates the unreliable motion vectors by interpolating them from the surrounding reliable motion vectors. In [32,65], median filtering and averaging are proposed for this purpose. The second approach re-estimates the motion vector on basis of the image intensities. The methods in [16, 54] exploit the correlation between pixels along the boundaries between adjacent image blocks. An erroneous motion vector is replaced by a new vector so that the mean squared difference in image intensity over the boundaries with the neighboring blocks is minimized. The approach in [47], which was developed in the context of blotch correction, re-estimates the motion of corrupted image blocks. The motion estimation process discards the corrupted pixels and constrains the smoothness of the motion vectors.

This section gives an indication of how well either approach can be expected to perform. Two algorithms are evaluated for this purpose. The first algorithm interpolates the unreliable motion estimates by applying the dilation interpolation technique described in Chapter 3 to the horizontal and vertical components of the motion vector fields independently. All weights $W_{m,n}$ are set to one, and no biases are applied.

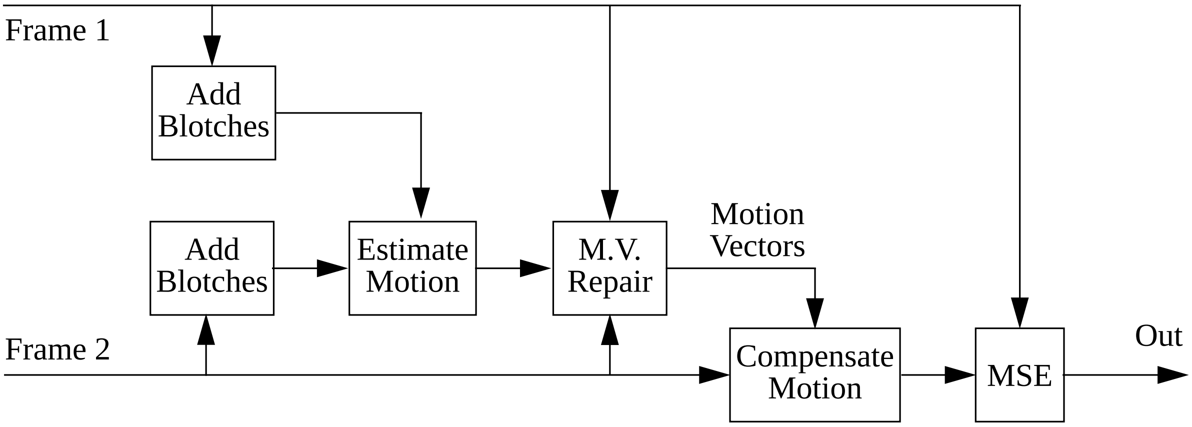


Figure 4.6 Scheme for measuring the effectiveness of motion vector repair.

The second algorithm re-estimates motion vectors on the basis of the observed image intensities, as illustrated by Figure 4.5. First, a bounding box is computed around each blotch with an additional (small) horizontal and vertical margin. Motion is estimated between the region contained by the box and the previous/next frames by block matching. To avoid biases resulting from blotched data, the block matcher discards the corrupted pixels in the current frame and in the reference frames. To limit the computational effort, the search range is limited to ±20 pixels for both horizontal and vertical directions.

Additionally, a third algorithm is evaluated. This algorithm simply replaces the motion vectors at blotched sites with vectors that indicate *zero motion*. Large parts of images tend to be temporally stationary and, therefore, assuming no motion is correct in a large number of cases.

The effectiveness of the three methods is evaluated by applying the scheme in Figure 4.6 to the same four test sequences used in the previous section. In this scheme, the motion vectors are estimated between pairs of consecutive frames that are corrupted by artificial blotches. Next, the motion vectors are repaired at locations indicated by the blotch detection masks. The blotch masks are made available to the motion vector repair block in Figure 4.6, though this is not shown explicitly in the figure. Then, one of the original, uncorrupted input frames is compensated for motion, and the MSE with the other original, uncorrupted input frame is computed. The MSE is computed only over the locations indicated by the blotch mask. If the motion vectors are accurate, the MSE will be small.

Two sets of experiments are carried out. The first set uses the true locations of the pixels corrupted by blotches. The second set uses detection masks resulting from the SDIa detector set to a detection rate of approximately 70%. The first set of experiments shows the improvements that are obtained under ideal circumstances. The second set of experiments shows the improvements obtained under realistic circumstances where false alarms influence the results. In this second set of experiments, 30% of the blotched pixels are not detected. These are the so-called *misses*. Motion vector repair does not influence the motion vectors assigned to misses because the motion vectors are only re-estimated at locations when blotches are detected.

Blotch Mask	Vector Repair	Western (MSE)	Mobcal (MSE)	Manege (MSE)	Tunnel (MSE)
Exact	None	94	338	1027	396
Exact	Block Matching	32	58	215	82
Exact	Dilation	62	218	656	122
Exact	Zero Motion	63	190	748	97
Estimated	None	157	721	2136	1044
Estimated	Block Matching	103	538	2138	693
Estimated	Dilation	140	594	2658	908
Estimated	Zero Motion	126	496	2841	847

Table 4.1 Evaluation of quality of motion vectors before and after motion vector repair. The MSE is computed only at sites indicated by the blotch mask. Both the true blotch mask and an estimated blotch mask, estimated with the SDIa detector set to a detection rate of approximately 70%, are used.

Blotch Mask	Western (MSE)	Mobcal (MSE)	Manege (MSE)	Tunnel (MSE)
Exact	21	35	113	18
Estimated	72	343	1925	689

Table 4.2 MSE computed with the motion vectors estimated from the original, unimpaired image sequences. The MSE is computed only at sites indicated by the blotch mask.

Table 4.1 gives the experimental results. This table indicates that applying vector repair significantly increases the accuracy of the corrupted motion vectors if the locations of the corrupted sites are known exactly. When the estimated blotch mask is used, the MSE increases and the gains are smaller. This is not surprising. Because of false alarms, motion vectors are re-estimated at locations that are not corrupted. The new motion estimates for the false alarms are suboptimal because correct image data are discarded in the motion estimation process.

The lowest MSEs are obtained with motion vectors repaired by the block matching technique, and, therefore, this method is to be preferred to the other methods for vector repair. The *zero motion* technique shows good results for those test sequences that contain large areas without motion, i.e., all test sequences except the *Manege* sequence. The dilation method has a relatively poor performance, yet it is to be preferred to no vector repair at all.

Table 4.2 shows the MSEs obtained from motion vectors computed from the original, unimpaired test sequences. These form the lower bound for the MSEs that can ideally be achieved. The conclusion is that the block-matching vector-repair technique bridges the gap between the MSE obtained from the corrupted vectors and the “true” vectors to a large extent.

4.2.3 Blotch correction techniques

MMF interpolator. A *multistage median filter* (MMF) is a concatenation of median filtering operations. The ML3Dex MMF is a heuristic method for interpolating missing data [49]. ML3Dex first applies five subfilters centered around the pixel being processed. Figure 4.7 shows the subfilter masks. In this figure, the top plane of each subfilter refers to data in the motion-compensated next frame, the center plane refers to data in the current frame, and the bottom plane refers to data in the motion-compensated previous frame. Next, the output of all the subfilters are combined and give the interpolated value according to:

$$m_l = median[W_l] \qquad \text{with } 1 \le l \le 5, \tag{4.15}$$

$$ML3Dex = median[m_1, m_2, m_3, m_4, m_5]. \tag{4.16}$$

Note that ML3Dex does not necessarily fulfill any of the image models used by the detectors described in Section 4.2.1. In other words, if a detector is applied again to a corrected image, the corrected data may well be flagged as being blotched. In such instances, there is no objective reason to prefer the corrected data to the observed data and, from an engineering point of view, it may actually be better to stick to the observed data. This reduces the risk of introducing corruption at locations at which blotches were mistakenly detected.

MRF interpolator. A MRF formulation towards interpolating missing data is given in [49]. This approach tries to find the MAP estimate of the missing data, $\hat{y}(\boldsymbol{i})$, given the locations of the corrupted sites and the observed (motion-compensated) previous, current and next frames by maximizing:

$$\begin{aligned} & P[\hat{y}(\boldsymbol{i}) | d(\boldsymbol{i}), z_{mc}(\boldsymbol{i}, t-1), z(\boldsymbol{i}), z_{mc}(\boldsymbol{i}, t+1)] \propto \\ & \exp\left(-\sum_{\boldsymbol{i}:d(\boldsymbol{i})=1} \left[\sum_{\boldsymbol{s} \in S_S(\boldsymbol{i})} (\hat{y}(\boldsymbol{i}) - \hat{y}(\boldsymbol{s}))^2 + \right.\right. \\ & \qquad \left.\left. \sum_{\boldsymbol{s} \in S_T(\boldsymbol{i})} \lambda \cdot [\hat{y}(\boldsymbol{i}) - z_{mc}(\boldsymbol{s}, t-1))^2 + (\hat{y}(\boldsymbol{i}) - z_{mc}(\boldsymbol{s}, t+1))^2]\right]\right), \end{aligned} \tag{4.17}$$

where S_S and S_T indicate the spatial and temporal neighborhoods, and λ is the relative weight for the temporal neighborhood. Equation (4.17) is optimized only over blotched image locations. The term $(\hat{y}(\boldsymbol{i}) - \hat{y}(\boldsymbol{s}))^2$ on the right hand side of (4.17) indicates the assumption that the interpolated values are likely to be smooth spatially. The other quadratic terms indicate the assumption that it is unlikely that the interpolated values introduce temporal discontinuities in image intensity along the motion trajectories. Equation (4.17) can be maximized with SA [28].

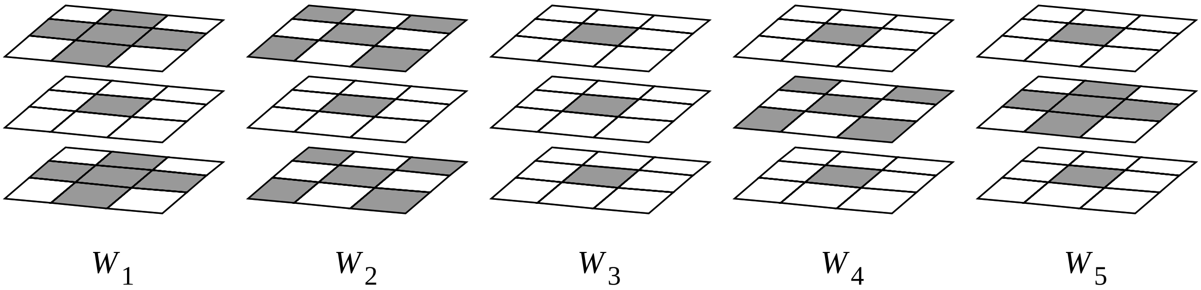


Figure 4.7 Subfilter masks for ML3Dex. Gray elements indicate the included data, white elements indicate excluded data.

AR interpolator.

A method for interpolating missing data based on a 3D AR model is described in [49]. For each region of an image with missing data a set of AR parameters is determined. It is assumed that the data in this region are stationary. The AR parameters are computed from data of the (motion-compensated) previous, current, and next frames. Note that the blotched data in the current frame t are discarded so that they do not bias the estimates of the AR parameters. Next, the missing data are interpolated so that the linear-mean-squared-prediction-error, computed with the estimated AR parameters, is minimized.

Consider the data to be ordered in a lexicographic fashion [74]. Let $\boldsymbol{e}$ indicate a vector of prediction errors, let $\boldsymbol{z}_+$ indicate a vector containing the observed data from the current frame plus that from the motion-compensated previous and next frames, and let $\boldsymbol{A}$ be a matrix with the AR coefficients placed at suitable locations. The prediction errors are denoted compactly by:

$$\boldsymbol{e} = \boldsymbol{A}\boldsymbol{z}_+. \tag{4.18}$$

The prediction errors consist of two parts. One part depends on the product of the known data $\boldsymbol{z}_{k+}$ (data that are not to be interpolated) in $\boldsymbol{z}_+$ with a number of columns from $\boldsymbol{A}$, these columns will be denoted by $\boldsymbol{A}_k$. The other part consists of the product of unknown data $\boldsymbol{z}_{u+}$ (data that are to be interpolated) in $\boldsymbol{z}_+$ and the remaining columns of $\boldsymbol{A}$, which will be denoted by $\boldsymbol{A}_u$:

$$\boldsymbol{e} = \boldsymbol{A}_k\boldsymbol{z}_{k+} + \boldsymbol{A}_u\boldsymbol{z}_{u+}. \tag{4.19}$$

The unknown data are interpolated so that the mean-squared-prediction-error $\boldsymbol{e}^T\boldsymbol{e}$ is minimized. Taking the derivative of $\boldsymbol{e}^T\boldsymbol{e}$ with respect to $\boldsymbol{z}_{u+}$, setting it to zero and solving for $\boldsymbol{z}_{u+}$ gives the required result:

$$z_{u+} = -[\boldsymbol{A}_u^T\boldsymbol{A}_u]^{-1}\boldsymbol{A}^T{}_u\boldsymbol{A}_k\boldsymbol{z}_{k+}. \tag{4.20}$$

Variations on this 3D AR method are described in [29, 45]. In [29] it is pointed out that the assumption of stationarity is not met for occluded regions that have become uncovered (and vice versa). The authors suggest estimating the AR model parameters and interpolating the missing data with two frames only. One frame is the current frame that contains the missing data. The other frame is either the preceeding or the following frame. This depends on which (motion-compensated) frame gives the smallest mean squared difference with the current frame in the region of the missing data. This method is referred to as the B3DAR method. In [45] this approach is refined by subdividing regions with missing data into multiple regions and interpolating the missing data for each region. This is done because a single set of AR coefficients may not be able to model a block of pixels adequately when the missing data cover a large region.

Drawbacks. There are a number of drawbacks to the methods for interpolating missing data described in this section. The multistage median filter has no model for the corrected image. Therefore the interpolation results are not necessarily consistent, either with the data surrounding the corrupted region or within the corrected region itself. The MRF interpolator gives an overly smooth result because it interpolates the data so that the differences between an interpolated pixel and its spatio-temporal neighbors are minimized. The MRF interpolator takes no measures for resolving the effects resulting from occlusion.

The AR interpolators can also smooth the data, and therefore the fidelity of the interpolated data in textured regions and in noisy film sequences is not that of their surroundings. As mentioned before, the problem of occlusion can, in principle, be solved with the method in [29]. However, unlike the method described in [29], the direction of interpolation should be determined pixelwise instead of blockwise. Because occlusion can vary on a pixel-by-pixel basis, the optimal direction of interpolation should be allowed to vary on a pixel-by-pixel basis. Furthermore, by subdividing missing data into a number of regions, as suggested in [45], mismatches may well occur within the interpolated results near the region boundaries.

Finally, all the approaches described in this section assume that the reference regions in the motion-compensated previous/next frames do not contain missing data in the regions of interest. This assumption is not always correct and can lead to incorrect interpolated data, as will be shown in Section 4.5.

4.2.4 Discussion

Existing techniques for blotch detection show good performance, though even better performance is desirable in an automated environment for image restoration. For example, consider the ROC curves in Figure 4.3. These indicate that the false alarm rate varies between 0.5 and 15% for a correct detection rate of 85%. With other words, not only are many blotches removed, which is good, but also two thousand to sixty thousand pixels are also interpolated unnecessarily for each frame of a PAL image, which has a resolution of 720×576 pixels. Because the interpolators are fallible, false alarms can lead to artifacts in the corrected sequence that are visually more disturbing than the blotches themselves.

The development of improved methods for blotch detection and correction is the topic of the remaining sections of this chapter. Sections 4.3 and 4.4 investigate how to improve the detectors. Section 4.5 develops an interpolator for correcting blotches that is robust to errors in the reference data obtained from motion-compensated frames.

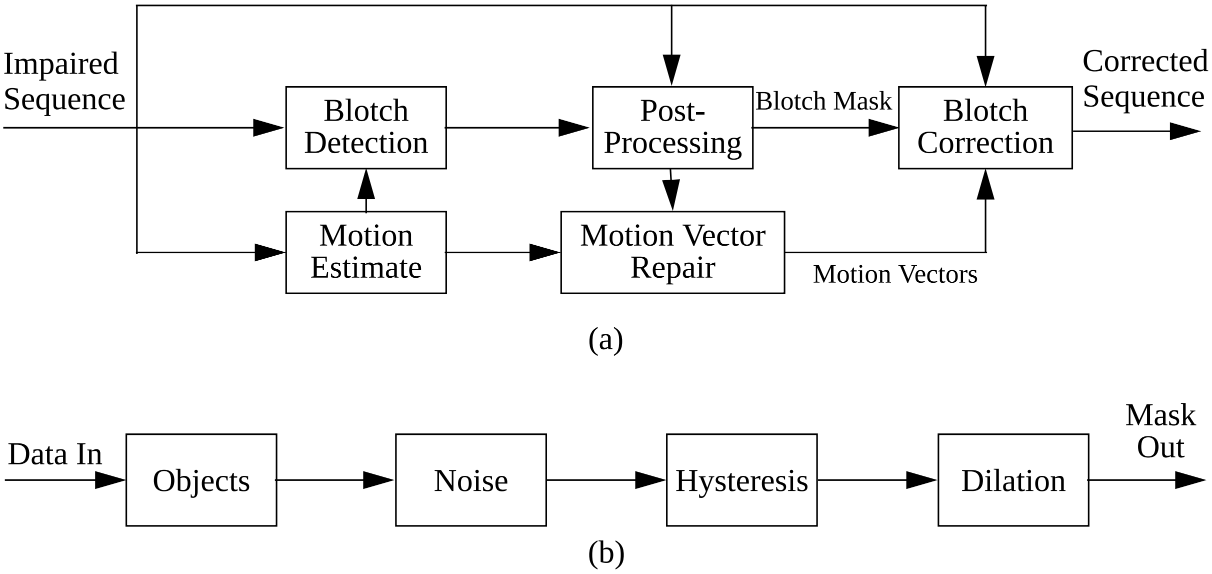


Figure 4.8 (a) Place of postprocessing in a system of blotch detection and correction, (b) chain of postprocessing operations for increasing the ratio between correct detections and false alarms.

4.3 Improved blotch detection by postprocessing

The goal of this section is to improve the ratio of correct detections to false alarms of existing blotch detectors. The approach taken here is not one of designing yet another detector. Instead, a strategy of postprocessing that removes possible false alarms and that finds parts of blotches missed by the detector is developed. Figure 4.8a shows how postprocessing fits in the scheme of Figure 4.1b. Figure 4.8b shows the proposed set of postprocessing techniques.

What is the idea behind the postprocessing operations? Blotches are not just random sets of individual pixels, but that they are spatially coherent regions and can be manipulated as such. How these regions can be extracted from the blotch detection masks is discussed in Section 4.3.2. Because it is not certain at this point that the extracted regions are true blotches, rather than something that resulted from false alarms made by the detector, the term *candidate blotches* is used to refer to the extracted regions.

Section 4.3.3 follows a probabilistic approach towards identifying and eliminating candidate blotches as a result of false alarms due to noise. The other candidate blotches, resulting from correct detections, have been detected only partially. Applying techniques called *hysteresis thresholding* and *constrained dilation* can make the detections more complete. These techniques are explained in Section 4.3.4 and Section 4.3.5. Section 4.3.6 concludes with experimental evaluations that demonstrate the effectiveness of the postprocessing approach applied to a simplified version of the ROD detector, which is described next.

4.3.1 Simplified ROD detector

By letting $T_2 \to \infty$ and $T_3 \to \infty$, the output of the ROD detector is completely determined by

T_1. In this case, T_2 and T_3 can removed from the equations and a *simplified ROD* (SROD) detector results. The SROD detector is computationally much more efficient than the ROD detector because it no longer requires the reference pixels to be ordered by rank:

$$SROD(\boldsymbol{i}) = \begin{cases} min(p_k) - z(\boldsymbol{i}) & \text{if } min(p_k) - z(\boldsymbol{i}) > 0 \\ z(\boldsymbol{i}) - max(p_k) & \text{if } z(\boldsymbol{i}) - max(p_k) > 0 \\ 0 & \text{otherwise} \end{cases} \quad \text{with } k = 1, ..., 6. \; . \qquad (4.21)$$

A blotch is detected if:

$$d_{SROD}(\boldsymbol{i}) = \begin{cases} 1 & \text{if } SROD(\boldsymbol{i}) > T_1 \\ 0 & \text{otherwise} \end{cases} \quad \text{with } T_1 \geq 0 \, . \qquad (4.22)$$

The SROD detector looks at a range of pixel intensities obtained from motion-compensated frames and compares this range to the pixel intensity under investigation. Blotches are detected if the current pixel intensity lies far enough outside the range. What is considered “far enough” is determined by T_1.

4.3.2 Extracting candidate blotches

The SROD detector is a pixel based detector. If the spatial coherence within blotches is to be exploited, regions consisting of pixels with similar properties will have to be extracted from the available data. Adjacent pixels within a blotch tend to have similar intensities. A pair of pixels are considered to be similar if their difference is smaller than twice the standard deviation of the noise. This means at least 96% of the pixels will be labeled as belonging to the same candidate blotch if additive white gaussian noise is assumed to be corrupting the image.

Therefore, adjacent pixels with similar intensities that have been flagged by the blotch detector are considered to be part of the same candidate blotch. To differentiate between the various candidate blotches, a unique label is assigned to each of them.

4.3.3 Removing false alarms due to noise

After the labelling procedure, a candidate blotch is an object with spatial support S and it consists of K pixels, each of which has a specific detector output $SROD(\boldsymbol{i})$. By selecting a small value for T_1, the detector is set to a great degree of sensitivity. In this case, it is not only sensitive to blotches, but also to noise. An example of this is given in Figure 4.9, which shows a frame from the original *Western* test sequence, the same frame degraded with artificial blotches, and the blotch mask used for adding the artificial blotches. The estimated blotch mask, estimated with the SROD detector with $T_1 = 0$, shows many false alarms.

Figure 4.9d also zooms in on a candidate blotch. The question for this candidate blotch is whether it is likely that it was detected purely as a result of false alarms due to noise. If so, the

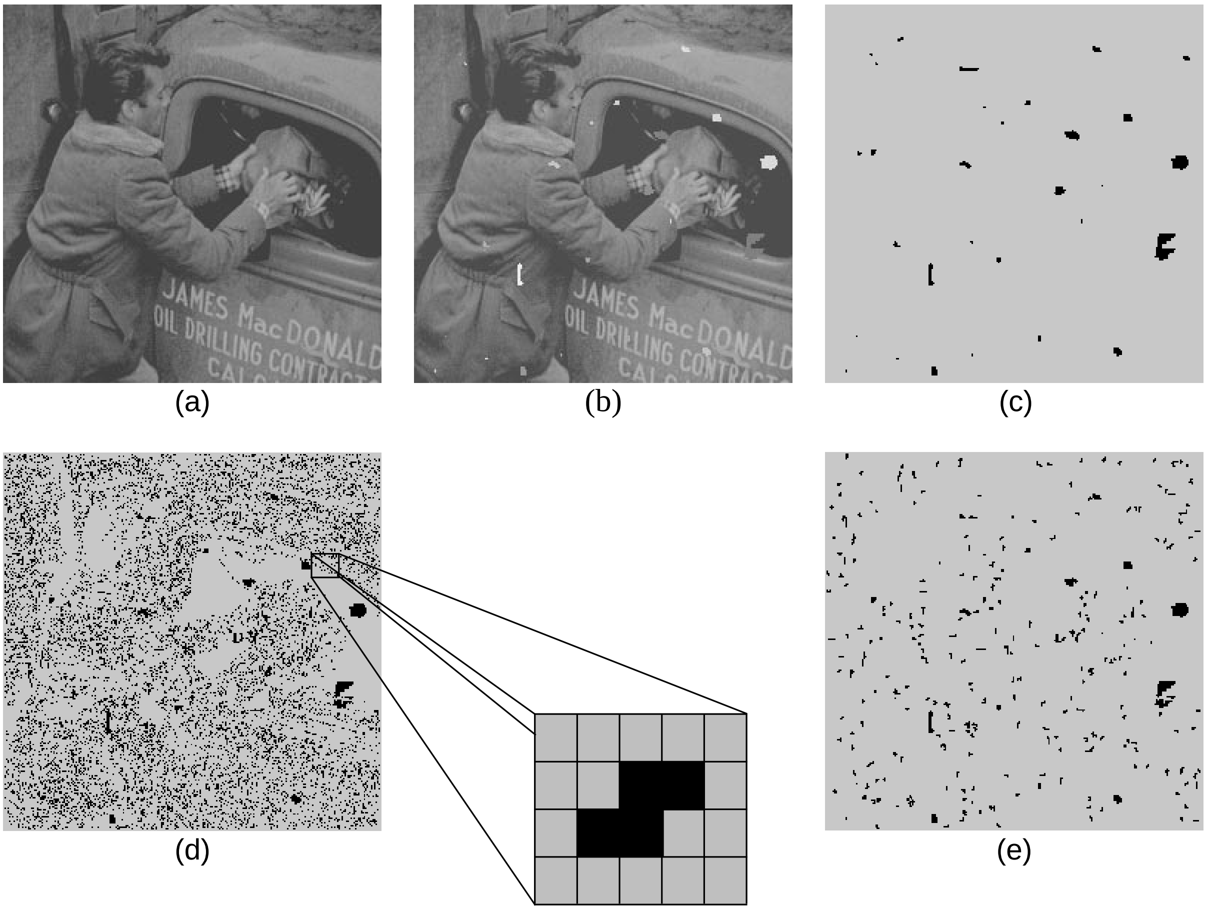


Figure 4.9 (a) Frame from Western test sequence, (b) same frame with artificially added blotches, (c) true blotch mask, (d) blotch mask estimated with the SROD detector with $T_1 = 0$ and a zoom in on a candidate blotch, (e) estimated blotch mask after possible false alarms due to noise are removed.

complete candidate blotch should be removed from the blotch detection mask. Figure 4.9(e) shows a result of this approach, for which the details follow, applied to Figure 4.9(d). Many false alarms have been removed.

The probability of a candidate blotch being detected purely due to false alarms is equal to the probability of the detector giving specific set of values $SROD(\boldsymbol{i})$, all of which are larger than T_1. This probability can be computed in two steps. The first step determines the probability of a specific detector response for an individual pixel under the influence of noise. The second step determines the probability that a collection of such pixels belong to a single object. The details of these two steps are given now.

For the first step, it is assumed that the reference pixels p_k and the current pixel $z(\boldsymbol{i})$ are identical except for the additive noise in the absence of blotches, i.e., $z(\boldsymbol{i}) = y + \eta_{\boldsymbol{i}}$ and $p_k = y + \eta_k$, where $\eta_{\boldsymbol{i}}$ and η_k indicate a specific noise realization. It is also assumed that the noise is i.i.d., has zero mean, and is symmetrically distributed around the mean. The probability that the SROD detector generates a false alarm due to noise is:

$$
\begin{aligned}
&P[SROD(\boldsymbol{i}) > T_1] \\
&= P[z(\boldsymbol{i}) - max(p_k) > 0, z(\boldsymbol{i}) - max(p_k) > T_1] + \\
&\quad P[min(p_k) - z(\boldsymbol{i}) > 0, min(p_k) - z(\boldsymbol{i}) > T_1] \\
&= P[z(\boldsymbol{i}) - max(p_k) > 0 | z(\boldsymbol{i}) - max(p_k) > T_1] \cdot P[z(\boldsymbol{i}) - max(p_k) > T_1] + \\
&\quad P[min(p_k) - z(\boldsymbol{i}) > 0 | min(p_k) - z(\boldsymbol{i}) > T_1] \cdot P[min(p_k) - z(\boldsymbol{i}) > T_1] \\
&= P[z(\boldsymbol{i}) - max(p_k) > T_1] + P[min(p_k) - z(\boldsymbol{i}) > T_1] \\
&= 2 \cdot P[z(\boldsymbol{i}) - max(p_k) > T_1] \\
&= 2 \cdot P[\eta_{\boldsymbol{i}} - max(\eta_k) > T_1],
\end{aligned}
\tag{4.23}
$$

where the last but one line follows from symmetry. Using the fact that $\eta_{\boldsymbol{i}} - max(\eta_k) > T_1$ requires that $\eta_{\boldsymbol{i}} - \eta_k > T_1$ for all k gives:

$$
\begin{aligned}
&P[SROD(\boldsymbol{i}) > T_1] \\
&= 2 \cdot P[\eta_{\boldsymbol{i}} - \eta_1 > T_1, \eta_{\boldsymbol{i}} - \eta_2 > T_1, \ldots, \eta_{\boldsymbol{i}} - \eta_6 > T_1] \\
&= 2 \int_{-\infty}^{\infty} P[\eta_{\boldsymbol{i}} - \eta_1 > T_1, \eta_{\boldsymbol{i}} - \eta_2 > T_1, \ldots, \eta_{\boldsymbol{i}} - \eta_6 > T_1 | \eta_{\boldsymbol{i}}] \cdot P[\eta_{\boldsymbol{i}}] \, d\eta_{\boldsymbol{i}} \\
&= 2 \int_{-\infty}^{\infty} \prod_k P[\eta_{\boldsymbol{i}} - \eta_k > T_1 | \eta_{\boldsymbol{i}}] \cdot P[\eta_{\boldsymbol{i}}] \, d\eta_{\boldsymbol{i}} \\
&= 2 \int_{-\infty}^{\infty} P^6[\eta_{\boldsymbol{i}} - \eta > T_1 | \eta_{\boldsymbol{i}}] \cdot P[\eta_{\boldsymbol{i}}] \, d\eta_{\boldsymbol{i}} \\
&= 2 \int_{-\infty}^{\infty} \left[\int_{-\infty}^{\eta(\boldsymbol{i}) - T_1} P[\eta] \, d\eta \right]^6 \cdot P[\eta_{\boldsymbol{i}}] \, d\eta_{\boldsymbol{i}}.
\end{aligned}
\tag{4.24}
$$

The step from the second to the third line in (4.24) is obtained by applying the theorem on total probability [55]. The fourth and the fifth lines in (4.24) are obtained by considering that the η_k are independent of each other. This is indicated by dropping index k from η_k. Equation (4.24) gives the probability that the SROD detector generates a false alarm for an individual pixel due to noise and can be evaluated numerically once the parameters of the noise have been determined.

In the case that the pixels of an image sequence are represented by integer values, the output of SROD also consists of integer values. The probability $P_{mass}[SROD(\boldsymbol{i}) = x]$ that the SROD detector gives a specific response x, with $x \geq 0$, for an individual pixel is given by:

$SROD(\boldsymbol{i})$	Probability	$SROD(\boldsymbol{i})$	Probability
1	0.091921	7	0.002224
2	0.060748	8	0.000892
3	0.036622	9	0.000304
4	0.020492	10	0.000108
5	0.010353	11	0.000028
6	0.005168	12	0.000007

Table 4.3 Probability of a specific detector response $SROD(\boldsymbol{i})$ computed for a constant signal corrupted by additive white gaussian noise with variance 9.6.

$$P_{mass}[SROD(\boldsymbol{i}) = x] = P[SROD(\boldsymbol{i}) > x - 1/2] - P[SROD(\boldsymbol{i}) > x + 1/2]. \qquad (4.25)$$

Table 4.3 lists the computed probabilities of specific detector responses in the case of white gaussian noise with a variance of 9.6. (The method described in [58] was used to estimate a noise variance of 9.6 for the *Western* sequence).

For the second step, it is assumed that the individual pixels within a blotch are flagged independently of their neighbors. Strictly speaking this assumption is incorrect because, depending on the motion vectors, sets of reference pixels p_k can overlap. The effects of correlation are ignored here. Let H_0 denote the hypothesis that an object is purely the result of false alarms and that each of the sets of reference pixels were identical to the true image intensity $y(\boldsymbol{i})$ except for the noise. $P[H_0]$ is then the probability that a collection of K individual pixels are flagged by the SROD detector, each of which with a specific response $x(\boldsymbol{i})$:

$$P[H_0] = \prod_{\boldsymbol{i} \in S} P_{mass}[SROD(\boldsymbol{i}) = x(\boldsymbol{i})], \qquad (4.26)$$

where S is the spatial support of the candidate blotch. Those objects for which the probability that they are solely the result of noise exceeds a risk R are removed from the detection mask:

$$P[H_0] > R. \qquad (4.27)$$

The result of this approach, as mentioned before, is indicated in Figure 4.9e.

4.3.4 Completing partially detected blotches

The technique for removing possible false alarms due to noise can be applied to any value of T_1. When a blotch detector is set to a low detection rate, not much gain is to be expected from this technique because the detector is insensitive to noise. A second method for improving the ratio of correct detections to false alarms is described here.

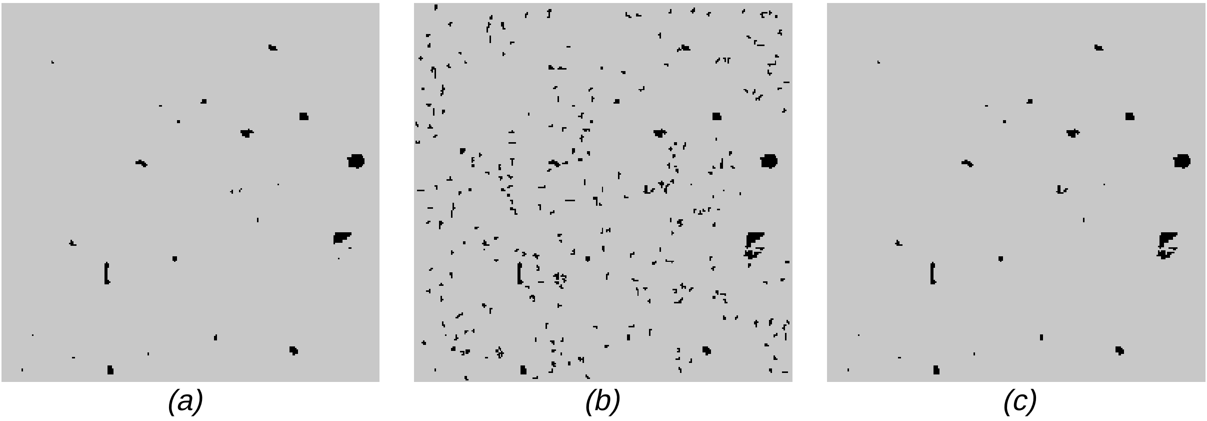

(a) (b) (c)

Figure 4.10 Example of hysteresis thresholding. Detection masks from (a) detector set to low sensitivity ($T_1 = 30$) with removal of possible false alarms due to noise, (b) detector set to high sensitivity ($T_1 = 0$) with removal of possible false alarms due to noise, (c) hysteresis thresholding.

Many blotches are not detected at all and others are detected only partially at lower detection rates. The strategy is now to make the partial detections more complete. This is achieved by noting from Figure 4.3 that the probability of false alarms decreases rapidly as the correct detection rate is lowered. Therefore, detections resulting from a blotch detector set to a low detection rate are more likely to be correct and can thus be used to validate the detections by the same detector set to a high detection rate.

The validation can be implemented by applying hysteresis thresholding [15]; see Figure 4.10. The first stage computes and labels the set of candidate blotches with a user-defined setting for T_1. Possible false alarms due to noise are removed as already described. The second stage sets the detector to a very high detection rate, i.e., $T_1 = 0$, and again a set of candidate blotches is computed and labeled. Candidate objects from the second set can now be validated; they are preserved if corresponding candidate objects in the first set exist. The other candidate blotches in the second set, which are more likely to have resulted from false alarms, are discarded. Effectively blotches detected with the operator settings are preserved and are made more complete.

4.3.5 Constrained dilation for missing details

There is always a probability that a detector fails to detect elements of a blotch, even when it is set to its most sensitive setting. For example, the large blotch on the right hand side in Figure 4.9c is not completely detected in Figure 4.9d. In this final postprocessing step, the detected blotches are refined by removing small *holes* in the candidate blotches and by adding parts of the blotches that may have been missed near the edges.

For this purpose, a constrained dilation operation is suggested here. Dilation is a well known technique in morphological image processing [57]. The constrained dilation presented here applies the following rule: if a pixel’s neighbor is flagged as being blotched and its intensity difference with that neighbor is small (e.g., less than twice the standard deviation of the noise) then that pixel should also be flagged as being blotched. The constraint on the differences in

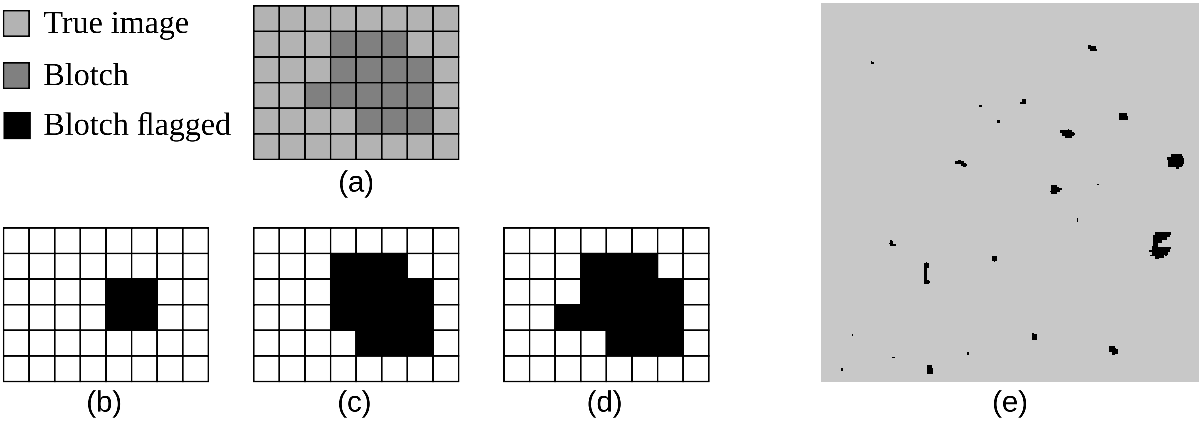


Figure 4.11 Example of constrained dilation: (a) image with blotches, (b) initial detection mask, (c) detection mask after one iteration, (d) detection mask after two iterations, (e) result of constrained dilation applied to Figure 4.10(c).

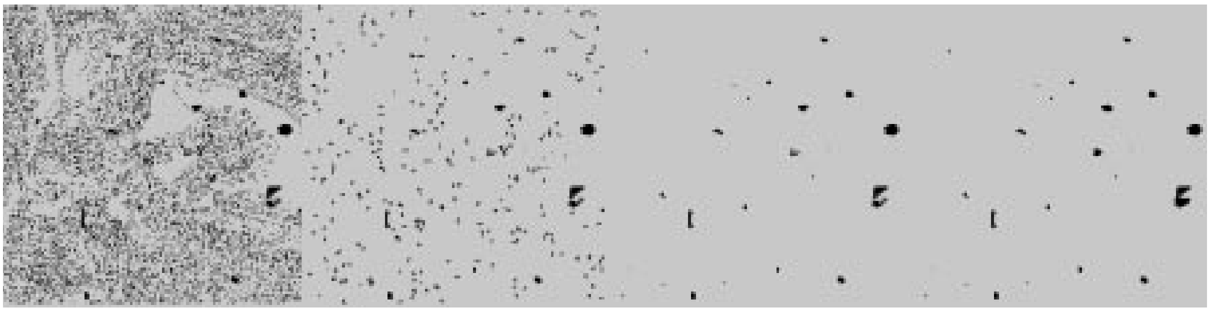
intensity reduces the probability that uncorrupted pixels surrounding a corrupted region are mistakenly flagged as being blotched. It uses the fact that blotches tend to have gray values that are significantly different from their surroundings. Figure 4.11a-c illustrates the procedure, Figure 4.11e shows the result of this method when applied to the blotch mask in Figure 4.10c.

It is important not to apply too many iterations of the constrained dilation operation because it is always possible that the contrast between a candidate blotch and its surrounding is small. The result would be that the candidate blotch grows completely out of its bounds and many false alarms occur. In practice, if the detector is set to a great sensitivity, applying two iterations favorably increases the ratio of the number of correct detections to false alarms. When the detector is set to less sensitivity, the constrained dilation is less successful and should not be applied. In the latter case, the blotches that are initially detected by the SROD detector must have sharp contrast with respect to the reference data. Because of the sharp contrast, the blotches are made fairly complete by the hysteresis thresholding. The dilation therefore adds little to the number of correct detections, yet it significantly increases the number of false alarms.

4.3.6 Experimental evaluation

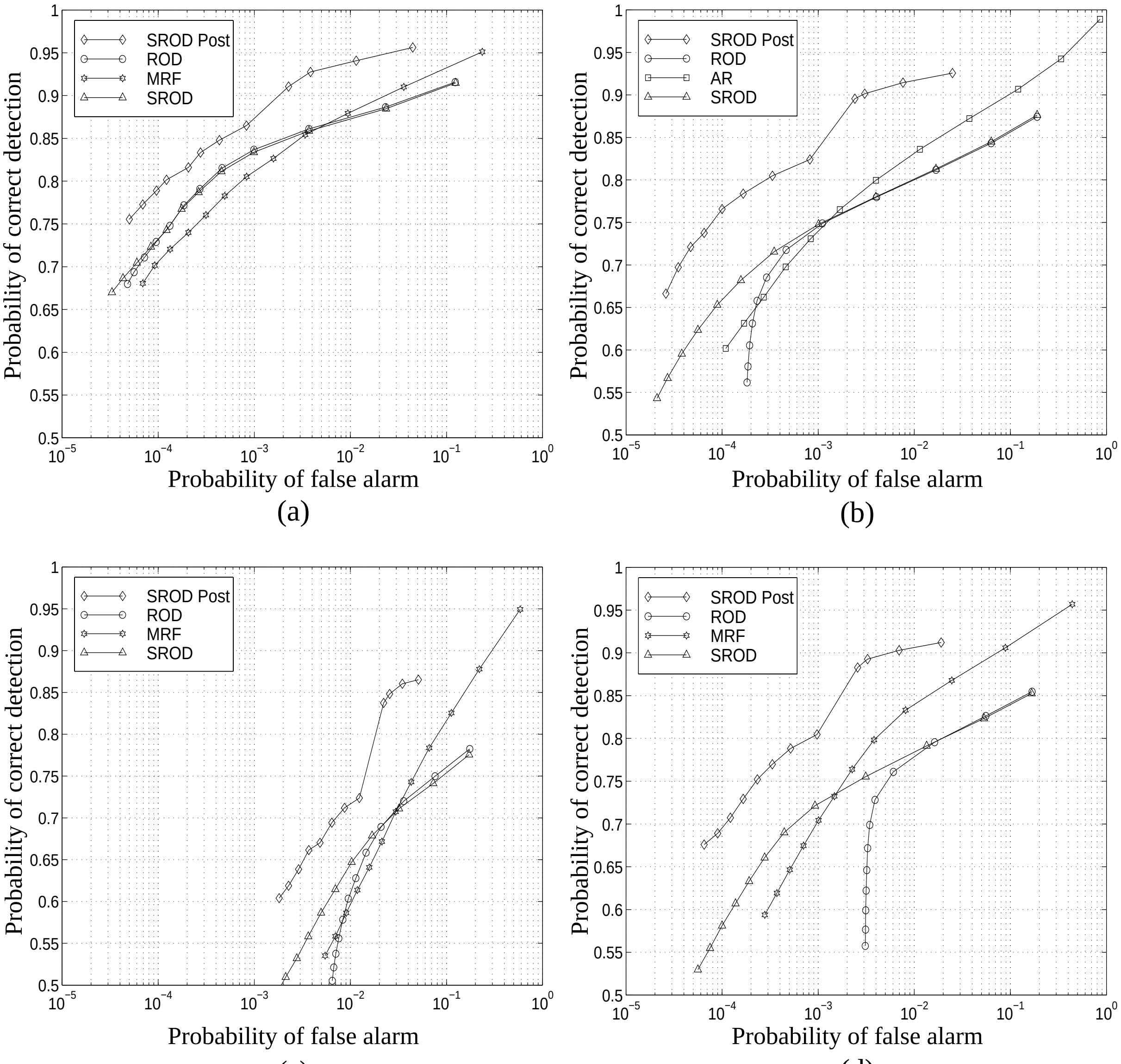
Figure 4.12 summarizes the effects of the consecutive postprocessing operations. Visually speaking, the final result in this figure compares well to the true blotch mask in Figure 4.9(c). Now the effectiveness of the postprocessing operations is evaluated objectively.

Figure 4.13 plots the ROCs for ROD detector, SROD detector, and the SROD detector with postprocessing. The results from either the MRF or the AR detector, depending on which showed the best results in Figure 4.3, are also plotted for comparison. Figure 4.13 makes it clear that the SROD detector has a performance similar to that of the ROD detector for small values of T_1 (high detection rates). When set to a lesser sensitivity, the SROD detector shows performance either similarly to or better than the ROD detector. This is explained by the fact



(a) (b) (c) (d)

Figure 4.12 Summary of postprocessing: (a) initial detection, (b) result after removal of false alarms, (c) result after hysteresis thresholding, (d) final result after constrained dilation.



(a) (b)

(c) (d)

Figure 4.13 Receiver operator characteristics for (a) Western sequence, (b) MobCal sequence, (c) Manege sequence, (d) Tunnel sequence.

that detection mask of the SROD detector is a subset of the detection mask of the ROD detector; each detection made by the SROD detector is also made by the ROD detector. However, the SROD detector generates not only fewer correct detections, but also (significantly) fewer false alarms.

The postprocessing applied to the detection masks obtained from the SROD detector improves the performance considerably over the whole range of operation of the detector. Note that the constrained dilation operation was not applied for $T_1 > 12$. This explains the sometimes large change in trend between the fourth and fifth measuring point of the SROD ROCs. The postprocessed results are significantly better than any results from the detectors without postprocessing. For instance, before postprocessing, a correct detection rate of 85% corresponds with a false alarm rate between 0.5 and 15%. After postprocessing a correct detection rate of 85% corresponds with a false alarm rate between 0.05 and 3%.

4.4 Blotch detection with increased temporal aperture

Objects for which the motion cannot be tracked accurately from frame to frame pose severe problems to blotch detectors. Incorrect motion vectors lead to incorrect sets of reference pixels and hence to false alarms. An obvious solution to this problem would be to use a “robust” motion estimator. Though techniques that are more robust to complex motion than the hierarchical block matcher used in this thesis do exist, e.g., motion estimators that use affine motion models [67,101], it is questionable whether the increase in performance justifies the increase in complexity. Motion in natural image sequences often involves objects of which shape, texture, illumination, and size vary in time. No motion estimation algorithm is truly capable of dealing with this type of motion.

An alternative way to reduce the number of false alarms is to incorporate more temporal information. False alarms result from the fact that object motion cannot be tracked to any of the reference frames. Increasing the number of reference frames increases the probability that good correspondence to at least one of the reference frames is found. Once good correspondence is found for an object, it is assumed that this object is not a blotch. Therefore, increasing the temporal aperture of a blotch detector reduces the number of false alarms. However, increasing the temporal aperture also increases the probability that blotches are mistakenly matched to other blotches or to some part of the image contents. This decreases the correct detection rate. Obviously there is a trade-off.

The SROD detector can easily be extended to use four reference frames by taking into account three extra reference pixels from each of the frames at $t - 2$ and at $t + 2$. The extended SROD detector is denoted by SRODex. The postprocessing operations can be applied as before, all that is necessary is to recompute the probability of false alarms due to noise (taking into account that there are now twelve reference pixels instead of six).

Consider two sets of candidate blotches detected by the SROD detector and the SRODex detector, respectively. The SRODex detections form a subset of the SROD detections; the SROD detector finds blotches everywhere the SRODex detector does, and more. The blotches detected by the SROD detector are more complete than those detected by the SRODex detector, but the SRODex detections are less prone to false alarms. As in Section 4.3.5, hysteresis

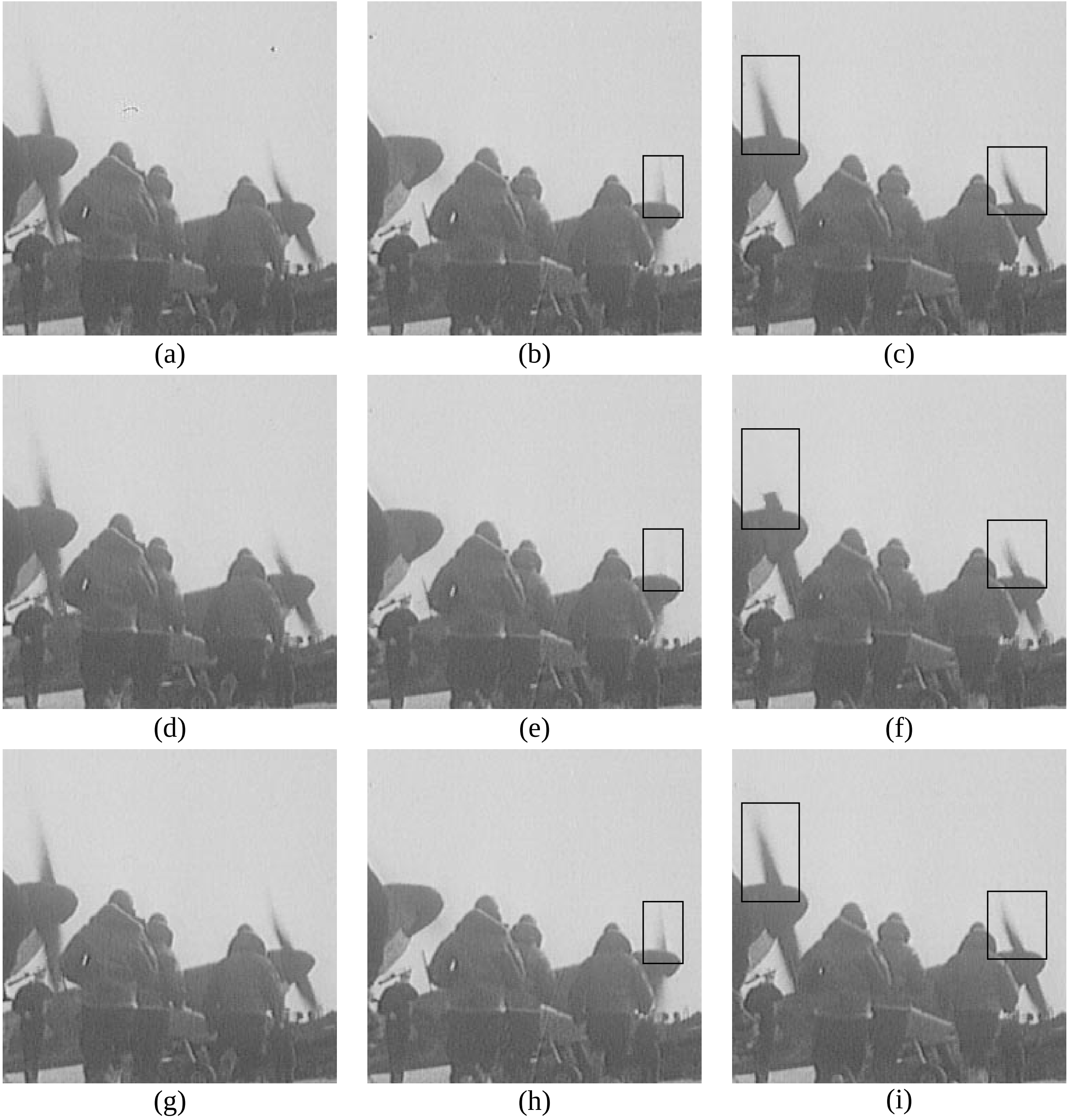


Figure 4.14 Top row: three consecutive frames from VJ Day sequence. Second row: corrected frames using SROD with postprocessing and ML3Dex. Note the distortion of the propellers in the boxed regions. Third row: corrected frames after combining the SROD detection results with SRODex results. (Original photos courtesy by the BBC).

thresholding can be applied. The reliable, possibly incomplete SRODex detected blotches can be used to validate less reliable, but more complete SROD detections. In case of true blotches, the shapes and sizes of the regions flagged by both detectors should be similar. If this is not the case, it is likely that the detections are a result of false alarms due to complex motion. Hence, preserving SROD-detected candidate blotches that are similar to corresponding SRODex-detected blotches reduces the probability of false alarms. The other SROD-detected candidate blotches are discarded. Two candidate blotches A and B are considered to be similar if the ratio of their sizes is smaller than some constant ζ:

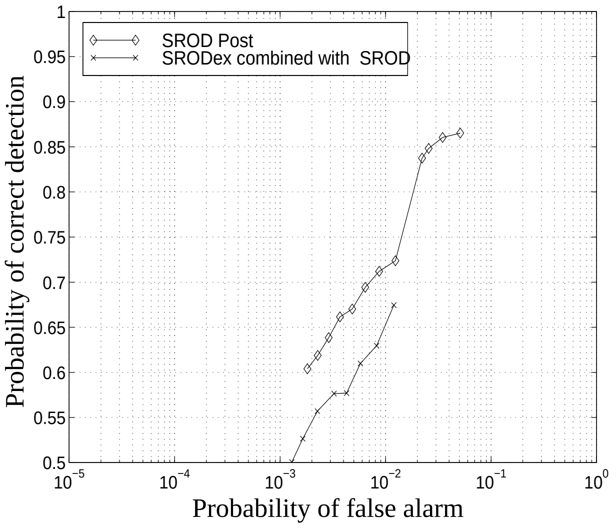


Figure 4.15 Receiver operator characteristics for the SROD detector and for the SROD detector combined with the SRODex detector (all with postprocessing) computed for the MobCal sequence.

$$\frac{\text{Size of blotch in A}}{\text{Size of blotch in B}} < \zeta . \tag{4.28}$$

Figure 4.14a-c show frames 27-29 from the *VJ Day* sequence. Besides blotches, this sequence contains a lot of action in the form running men and rotating propellers. Some of the propellers are not visible at all in some of the frames. Figure 4.14d-f shows data restored from the SROD detector ($T_1 = 10$) with postprocessing and ML3Dex for interpolation. The blotches have been removed very efficiently, but, as an unwanted side effect, parts of the propellers have been removed as well. Figure 4.14g-i shows restored data, but now the proposed combination of SROD and SRODex has been used with $T_1 = 10$ and $\zeta = 2$. Most of the blotches have been removed and, very importantly, the propellers have been preserved.

The proposed algorithm was very successful for the *VJ Day* sequence because it is capable of dealing with the periodic presence of the propellers. However, increasing the temporal aperture does not necessarily always increase the performance, as can be observed from the ROC curves for the *Manege* sequence in Figure 4.15. In this case, the SRODex detector misses too many correctly SROD-detected are discarded. Whether increasing the temporal aperture is beneficial to the restoration process depends on the particular image sequence. In practice, it is up to an operator to decide which detector is most appropriate.

4.5 Fast, good quality interpolation of missing data

Section 4.2.1 showed that model-based interpolation of missing data can be done with 3D AR processes. This method gives good-quality interpolation results and its performance in resynthesizing textures of missing data is superior to that of other interpolators. Equation (4.20) gives a closed form solution to the 3D AR interpolation method. Unfortunately, there are a number of drawbacks to this method. First, it is very expensive in computational terms. For example, resynthesizing the texture for a region with a blotch of 20×20 pixels requires inverting a matrix with 400×400 elements. Second, the method as described in Section 4.2.3 assumes that the data in the reference frames are always correct. This assumption is not always true. Incorrect reference data can result from erroneous motion estimates, occlusions, and corruptions due to blotches. Third, AR interpolations can be overly smooth if the interpolated regions are large.

It is important to realize that full 3D AR restoration is not necessary in most cases. The most common differences between the frames of an image sequence can be characterized by a rearrangement of the object location. Therefore, it is likely that missing data in one frame can be restored by pasting (copying) pixels from corresponding regions in a reference frame. Reliable motion estimates must be available for pasting. In fact, pasting can be viewed as a one-tap AR interpolator with a coefficient of 1.0.

This section investigates the concept of interpolating missing data by pasting. Each pixel of the missing data in a blotched frame is replaced by a pixel from the corresponding location in either the motion-compensated previous frame or the motion-compensated next frame. A strategy for determining the direction of interpolation (i.e., pasting from the previous frame or pasting from the next frame) is required. The strategy used here constrains the interpolated data to fit in well with the region surrounding the missing data. Hence, the data surrounding the missing data define a set of boundary conditions to the solution of the interpolation problem. This constraint is enforced by requiring corrected image regions to follow 2D AR processes as well as possible.

The question is now how to decide which reference frame should supply the pixels for pasting. One approach is to paste complete regions from either the previous or the next frame, depending on which result fits in better according to the 2D AR process (Figure 4.16a). To get good visual results with this approach, the motion-compensated reference data must represent the missing data at all locations. This requirement is less likely to be fulfilled as the size of the region to be pasted increases. The probability of some of the missing data being unavailable is proportional to the size of the region due possible to occlusions, blotches, and erroneous motion estimates.

A better approach is to determine the direction of interpolation for each individual pixel, as illustrated by Figure 4.16b. The pixel intensity from the motion-compensated reference frame with the smallest prediction error is pasted into the current frame. The advantage of pasting single pixels from either the previous or next reference frame is evident: if the reference data in one reference frame are inconsistent with the 2D AR model for the corrected frame, large prediction errors will result. In which case, data can be pasted from the other reference frame. This mechanism requires no explicit knowledge about errors in the reference data. Hence, corruptions in one of the reference frames do not influence the interpolated result negatively if the data in the other reference frame are correct.

At this point, the direction of interpolation in the pasting method described can vary erratically from pixel to pixel. Everything depends on which reference frame provides the pixel closest to the value predicted by the AR model. This can lead to two possible side effects. First, AR predictors tend to give overly smooth prediction results. Because the reference pixels closest to the AR predictions are selected, the pasted result can be overly smooth. Second, if the textures in reference frames are different (e.g., due to uncovering/occlusion), the pasted result might be a mixture of textures. In this case, the result is different from the true texture that underlies the missing data. These effects can be avoided by constraining the direction of interpolation to be consistent locally. For this purpose, a markov random field is applied.

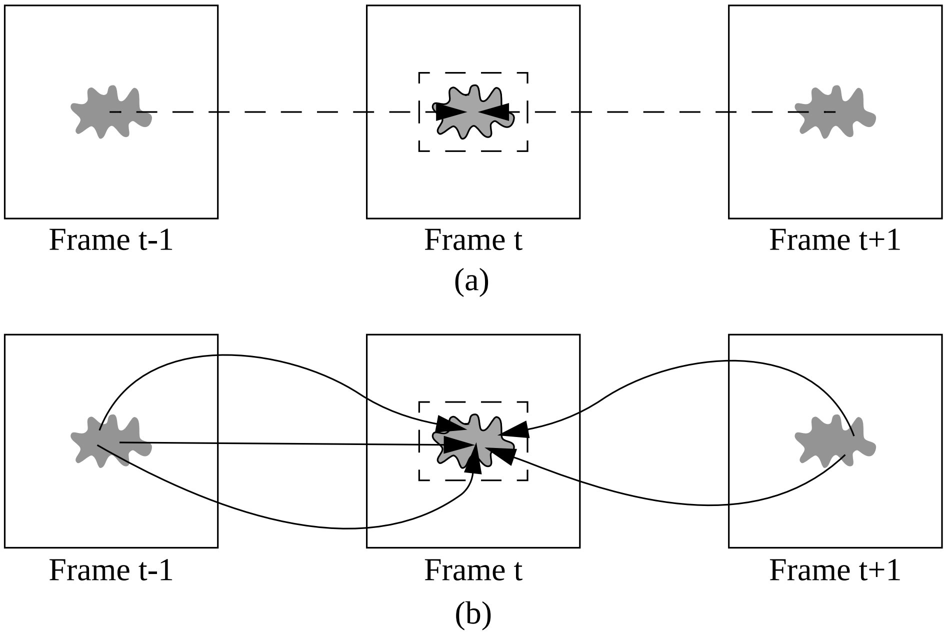


Figure 4.16 (a) Region-based pasting: a region from either the previous or next frame (motion-compensated) is pasted into the current frame, (b) pixel-based pasting: pixels are pasted from either of the reference frames. In both cases (a) and (b), the pasting is done so that the corrected region, indicated by the dashed box, can fit a 2D AR model as well as possible.

4.5.1 Interpolating missing data with controlled pasting

This section formulates the ideas mentioned in mathematical terms. Because the aim is to paste pixels from either the previous or the next motion-compensated frame, a binary *direction mask* $o(\boldsymbol{i})$ is introduced. This mask indicates for each spatial location which of the motion-compensated reference frames is most appropriate to serve as a reference for pasting, e.g., “0” for $z_{mc}(\boldsymbol{i}, t-1)$ and “1” for $z_{mc}(\boldsymbol{i}, t+1)$.

At this point it is assumed that the binary blotch detection mask $d(\boldsymbol{i})$ has already been determined. This could be done by any of the methods in the previous sections. The corrected frame $\hat{y}(\boldsymbol{i})$, which is an estimate of the true data $y(\boldsymbol{i})$, is given by:

$$\hat{y}(\boldsymbol{i}) = \begin{cases} z_{mc}(\boldsymbol{i}, t-1) & \text{if } d(\boldsymbol{i}) = 1, o(\boldsymbol{i}) = 0 \\ z_{mc}(\boldsymbol{i}, t+1) & \text{if } d(\boldsymbol{i}) = 1, o(\boldsymbol{i}) = 1. \\ z(\boldsymbol{i}) & \text{otherwise} \end{cases} \tag{4.29}$$

Now, the aim is to find $o(\boldsymbol{i})$. The reconstructed image $\hat{y}(\boldsymbol{i})$ follows through this variable. The image data model underlying the corrected image $\hat{y}(\boldsymbol{i})$ is assumed to be a 2D AR model of order n with coefficients a_l, with $l = 1, \ldots, n$. The prediction error $e(\boldsymbol{i})$ is a gaussian random variable with zero mean and variance σ_e^2:

$$\hat{y}(\boldsymbol{i}) = \sum_{l=1}^{n} a_l \cdot \hat{y}(\boldsymbol{i} + \boldsymbol{q}_l) + e(\boldsymbol{i}). \tag{4.30}$$

The binary field $o(\boldsymbol{i})$ must be found so that, on one hand, the corrected image $\hat{y}(\boldsymbol{i})$ fits the image model in (4.30) as well as possible, i.e., so that the prediction error variance σ_e^2 is as low as possible. On the other hand, as already explained, the direction of interpolation must be a consistent one locally. Note that not only must $o(\boldsymbol{i})$ be found, but also the parameters that define the AR process, namely the AR coefficients a_l and the prediction error variance σ_e^2.

To come to a tractable solution, the number of computations must be kept as low as possible. Therefore $o(\boldsymbol{i})$ is not computed for the complete frame. Instead, $o(\boldsymbol{i})$ is computed only for regions that contain missing data. The image regions are selected so that, at most, 20% of the area consists of missing data. Each region is modeled by a single set of AR model parameters a_l and a single prediction error variance σ_e^2.

Proceeding in a probabilistic fashion, these requirements translate to finding the maximum of $P[o(\boldsymbol{i}), a_1, \ldots, a_n, \sigma_e^2 | z_{mc}(\boldsymbol{i}, t-1), z(\boldsymbol{i}), z_{mc}(\boldsymbol{i}, t+1), d(\boldsymbol{i}), O]$. Here O indicates the direction of interpolation for the pixels in the local region surrounding $o(\boldsymbol{i})$. With Bayes' rule, this can be seen to be proportional to:

$$P[o(\boldsymbol{i}), \boldsymbol{a}, \sigma_e^2 | \boldsymbol{z}_+(\boldsymbol{i}), d(\boldsymbol{i}), O] \propto P[\boldsymbol{z}_+(\boldsymbol{i}) | o(\boldsymbol{i}), \boldsymbol{a}, \sigma_e^2, d(\boldsymbol{i})] \cdot P[o(\boldsymbol{i}) | O] \cdot P[\boldsymbol{a}] \cdot P[\sigma_e^2], \tag{4.31}$$

where the terms $a_1, \ldots, a_n$ have been grouped together into $\boldsymbol{a}$ and $\boldsymbol{z}_{mc}(\boldsymbol{i}, t-1), \boldsymbol{z}(\boldsymbol{i})$, and $\boldsymbol{z}_{mc}(\boldsymbol{i}, t+1)$ have been grouped together in $\boldsymbol{z}_+(\boldsymbol{i})$ for convenience.

The first term on the right hand side of (4.31), $P[\boldsymbol{z}_+(\boldsymbol{i}) | o(\boldsymbol{i}), \boldsymbol{a}, \sigma_e^2, d(\boldsymbol{i})]$, indicates the likelihood of observing the data $\boldsymbol{z}_+(\boldsymbol{i})$, given the direction of interpolation, the AR model parameters, and the blotch mask. Let $AR(\hat{y}, \boldsymbol{a}, \boldsymbol{i})$ be the prediction of the corrected image $\hat{y}$ at location $\boldsymbol{i}$. $AR(\hat{y}, \boldsymbol{a}, \boldsymbol{i})$ is determined completely by $\boldsymbol{z}_+(\boldsymbol{i}), o(\boldsymbol{i}), \boldsymbol{a}, \sigma_e^2$ and $d(\boldsymbol{i})$. The likelihood can then be defined by (4.32).

$$\begin{aligned} & P[\boldsymbol{z}_+(\boldsymbol{i}) | o(\boldsymbol{i}), \boldsymbol{a}, \sigma_e^2, d(\boldsymbol{i})] \propto \\ & \exp(-[(1 - d(\boldsymbol{i})) \cdot \frac{(z(\boldsymbol{i}) - AR(\hat{y}, \boldsymbol{a}, \boldsymbol{i}))^2}{2\sigma_e^2} + \\ & \quad d(\boldsymbol{i}) \cdot o(\boldsymbol{i}) \cdot \frac{(z_{mc}(\boldsymbol{i}, t+1) - AR(\hat{y}, \boldsymbol{a}, \boldsymbol{i}))^2}{2\sigma_e^2} + \\ & \quad d(\boldsymbol{i}) \cdot (1 - o(\boldsymbol{i})) \cdot \frac{(z_{mc}(\boldsymbol{i}, t-1) - AR(\hat{y}, \boldsymbol{a}, \boldsymbol{i}))^2}{2\sigma_e^2})]. \end{aligned} \tag{4.32}$$

The second line in (4.32) states that at locations at which no blotches have been detected ($d(\boldsymbol{i}) = 0$), the likelihood of observing a specific pixel intensity in the current frame $z(\boldsymbol{i})$ is proportional to the squared AR prediction error weighted by the prediction error variance. The third line in (4.32) states that, at locations where a pixel from $z_{mc}(\boldsymbol{i}, t+1)$ was pasted, the likelihood of that pixel intensity being observed is proportional to the weighted squared AR prediction error of the restored frame. The fourth line of (4.32) makes a statement somewhat similar to that in the third line, but then for $z_{mc}(\boldsymbol{i}, t-1)$. Equation (4.32) can be simplified to:

$$\begin{aligned} & P[\boldsymbol{z}_{+}(\boldsymbol{i}) | o(\boldsymbol{i}), \boldsymbol{a}, \sigma_e^2, d(\boldsymbol{i})] \\ & \propto \exp\left(-\frac{((1-d(\boldsymbol{i}))\cdot z(\boldsymbol{i}) + d(\boldsymbol{i})\cdot(o(\boldsymbol{i})\cdot z_{mc}(\boldsymbol{i}, t+1) + (1-o(\boldsymbol{i}))\cdot z_{mc}(\boldsymbol{i}, t-1)) - AR(\hat{y}, \boldsymbol{a}, \boldsymbol{i}))^2}{2\sigma_e^2}\right) \\ & \propto \exp\left(-\frac{(\hat{y}(\boldsymbol{i}) - AR(\hat{y}, \boldsymbol{a}, \boldsymbol{i}))^2}{2\sigma_e^2}\right) \\ & \propto \frac{1}{\sqrt{2\pi\sigma_e^2}} \exp\left(-\frac{e(\boldsymbol{i})^2}{2\sigma_e^2}\right). \end{aligned} \tag{4.33}$$

This means that likelihood function of the observed data $P[\boldsymbol{z}_{+}(\boldsymbol{i}) | \ldots]$ is proportional to probability of the prediction error $e(\boldsymbol{i})$ of the restored frame as defined by (4.30).

The other three terms in (4.31) describe a priori knowledge related to the model parameters. To achieve local consistency in the direction mask $o(\boldsymbol{i})$, the following prior is assumed:

$$P[o(\boldsymbol{i}) | O] \propto \exp\left(-\sum_k \beta \left|o(\boldsymbol{i}) - o(\boldsymbol{i} + \boldsymbol{q}_k)\right|\right), \tag{4.34}$$

where β is a constant that defines the strength of the self-organization. The eight-connected neighbors of $o(\boldsymbol{i})$ are indicated by $o(\boldsymbol{i} + \boldsymbol{q}_k)$, with $k = 1, \ldots, 8$. Equation (4.34) simply states that the direction of interpolation for a pixel is likely to be similar to that of the majority of its neighbors.

Following [47], a uniform prior is assigned to **a**, and a Jeffreys' prior [68] is assigned to the prediction error variance σ_e^2:

$$P[\sigma_e^2] \propto 1/\sigma_e^2. \tag{4.35}$$

Equation (4.31) is completely defined now. The next section describes the practical implementation for correcting blotched image sequences on the basis of maximizing (4.31) jointly for all $o(\boldsymbol{i})$ in a region with missing data.

4.5.2 Practical implementation of controlled pasting

The MAP estimate for (4.31) jointly for all $o(\boldsymbol{i})$ can be found with SA [28]. SA as described here involves two elements. The first element is a global control parameter T called *temperature*, which is used to shape the probability functions in (4.31). The second element is a mechanism for drawing random samples from conditionals, called a *Gibbs sampler*. SA can be summarized by four steps:

1. Initialize temperature: $T = T_{begin}$,
2. Sample the unknowns with the Gibbs sampler,
3. Repeat step 2 until convergence is obtained,
4. Lower T according to a cooling schedule and go to 2 if $T > T_{final}$.

In [28] it is proved that if T_{begin} is sufficiently large and that if a logarithmic cooling schedule is applied, the algorithm converges to the MAP solution. The most involved part of the SA scheme is the Gibbs sampler. The Gibbs sampler operates iteratively by drawing random samples for the unknowns in turn, which are derived in Appendix B:

$$\begin{aligned}
&\boldsymbol{a} \sim P[\boldsymbol{a}|\sigma_e^2, o(\boldsymbol{i}), z_+, d],\\
&\sigma_e^2 \sim P[\sigma_e^2\big|\boldsymbol{a}, o(\boldsymbol{i}), z_+, d],\\
&o(i) \sim P[o(i)|\boldsymbol{a}, \sigma_e^2, z_+(\boldsymbol{i}), d(\boldsymbol{i}), O].
\end{aligned} \tag{4.36}$$

One might argue that using such heavy machinery as SA just to determine the direction of interpolation for a set of pixels is slightly overdoing things. The goal of this section is to simplify this machinery somewhat and to come to an efficient implementation.

The number of computations has to be kept small for an efficient implementation. As mentioned in the previous section, the controlled pasting scheme is not applied to the complete image, but only to image regions containing missing data. The image regions are selected so that, at most, 20% of the area consists of missing data. A single set of three AR model parameters a_l is computed for each region. A quarter plane prediction model is used (see Figure 4.17).

Strictly speaking, all unknowns should be sampled in the SA scheme, and this includes the sampling the AR coefficients $\boldsymbol{a}$ and the error variance σ_e^2 from the probability functions

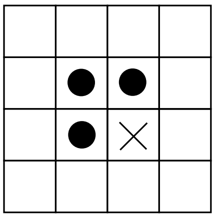


Figure 4.17 Support (dots) used for AR prediction (cross).

derived in Appendix B. Drawing samples from these distributions is costly in terms of computational complexity, and it is noted here that good results are obtained by just using the least squares estimate for the AR coefficients instead of sampling them. (In fact, this is equivalent to sampling from (B.10) with zero variance). Similarly, it is not necessary to sample for σ_e^2 to get good results. Hence:

$$\boldsymbol{a} = \boldsymbol{R}_{\hat{y}\hat{y}}^{-1} \cdot \boldsymbol{r}_{\hat{y}\hat{y}}. \tag{4.37}$$

Here $\boldsymbol{R}_{\hat{y}\hat{y}}$ and $\boldsymbol{r}_{\hat{y}\hat{y}}$ are the autocorrelation matrix and autocorrelation vector that are required for solving the normal equations [53,94]. What remains are the samples to be drawn for $o(i)$ from (B.16):

$$\begin{aligned} &P[o(i)|\boldsymbol{a}, \sigma_e^2, z_+(\boldsymbol{i}), d(\boldsymbol{i}), O] \\ &\propto \exp\Bigg(-\frac{1}{T}[(1-d(\boldsymbol{i}))\cdot(z(\boldsymbol{i})-AR(\hat{y},\boldsymbol{a},\boldsymbol{i}))^2 + \\ &\qquad d(\boldsymbol{i})\cdot(o(\boldsymbol{i})\cdot z_{mc}(\boldsymbol{i},t+1)+(1-o(\boldsymbol{i}))\cdot z_{mc}(\boldsymbol{i},t+1)-AR(\hat{y},\boldsymbol{a},\boldsymbol{i}))^2 + \\ &\qquad \sum_k \beta\left|o(\boldsymbol{i})-o(\boldsymbol{i}+q_k)\right|]\Bigg), \end{aligned} \tag{4.38}$$

where $AR(\hat{y}, \boldsymbol{a}, \boldsymbol{i})$ indicates the spatial AR prediction of $\hat{y}(\boldsymbol{i})$ from its surroundings. The reconstructed image $\hat{y}$, required for the AR predictions is obtained via (4.29). Drawing samples from (4.38) with the Gibbs sampler is very easy. It involves evaluating (4.38) at a specific site $\boldsymbol{i}$ for $o(\boldsymbol{i}) = 0$ and for $o(\boldsymbol{i}) = 1$, while keeping the other values for the direction mask and the $\hat{y}(\boldsymbol{i})$ fixed. The results are assigned to c_1 and c_2, respectively. Next a value for $o(\boldsymbol{i})$ (and thereby the corresponding $\hat{y}(\boldsymbol{i})$) is chosen at random, with a probability $c_1/(c_1+c_2)$ that $o(\boldsymbol{i}) = 0$ and with a probability $c_2/(c_1+c_2)$ that $o(\boldsymbol{i}) = 1$. A single update of an image region consists of applying the Gibbs sampler to each site in that region in turn, using, for instance, a checkerboard scanning pattern.

Figure 4.18 summarizes the practical *controlled pasting* (CP) scheme that results. The data put into the system consist of the current frame and the motion-compensated previous and next frames. The blotch detection mask, which indicates for each pixel whether it is considered to be part of a blotch, also belongs to the input data. Initially, the direction field $o(\boldsymbol{i})$ is assigned binary values at random, and an initial temperature T is chosen. The main loop is as follows. First a corrected frame $\hat{y}(\boldsymbol{i})$ is generated. Next, a set of AR coefficients $\boldsymbol{a}$ is estimated for each missing region. This is used for predicting the corrected image intensities. Next, the direction of interpolation is updated by sampling from (4.38) as already described. The main loop is repeated at each temperature level T, until the solution has converged or until a fixed number of iterations have been done. The temperature is lowered with an exponential cooling schedule:

$$T_k = \gamma^k \cdot T_{begin}, \tag{4.39}$$

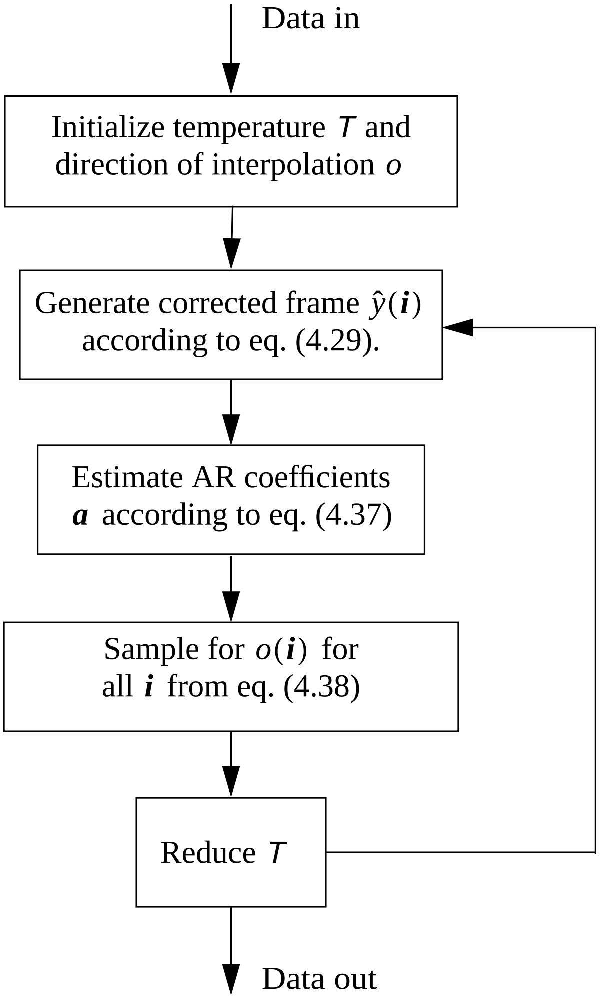

Data in
Initialize temperature T and direction of interpolation o
Generate corrected frame $\hat{y}(\boldsymbol{i})$ according to eq. (4.29).
Estimate AR coefficients $\boldsymbol{a}$ according to eq. (4.37)
Sample for $o(\boldsymbol{i})$ for all $\boldsymbol{i}$ from eq. (4.38)
Reduce T
Data out

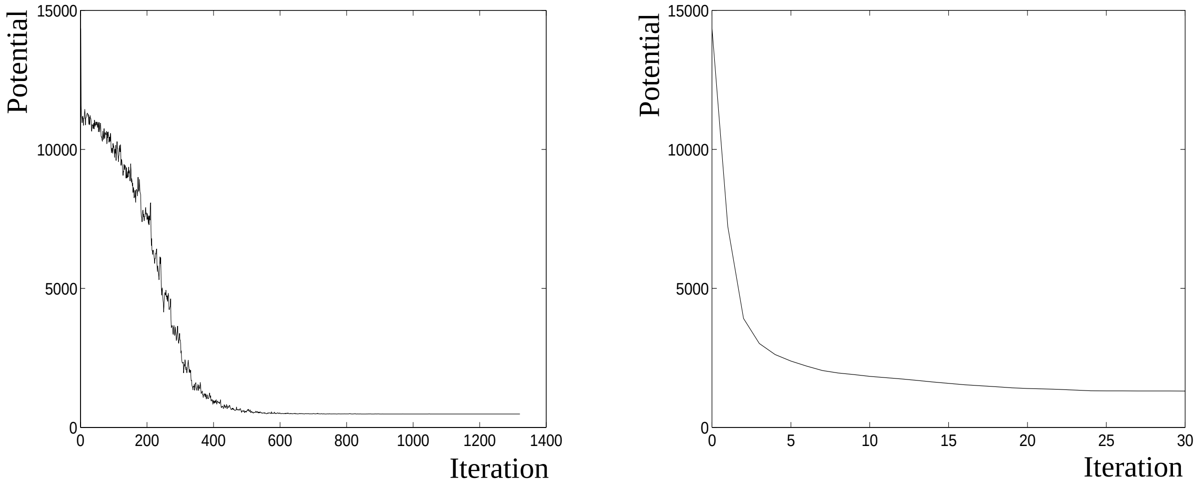
Figure 4.18 Overview of a practical implementation of the CP scheme.

where γ controls the rate of decrease and k indicates the kth temperature level. The main loop is iterated again until the final temperature has been reached.

4.5.3 Experiments with controlled pasting

The scheme in Figure 4.18 is ready to be applied now. The result it yields is the joint distribution of the $o(\boldsymbol{i})$ within an image region S_r, as is given by (4.40). The term defined by the summation in (4.40) is known as the *potential function*. Lower potential functions indicate better solutions.

To get some idea about what sensible values are for T_{begin}, T_{final}, and κ, two experiments are carried out on a blotched frame from the *Western* test sequence. For the first experiment $T_{begin} = 100.0$, $T_{final} = 1.0$ and $\gamma = 0.9$ is chosen. At each temperature level, 30 iterations are applied. For the second experiment, only one temperature level $T_{begin} = T_{final} = 1$ is assumed. Again, 30 iterations of the Gibbs sampler are applied. Figure 4.19 plots the potential functions for both experiments as a function of the number of iterations.



(a) (b)

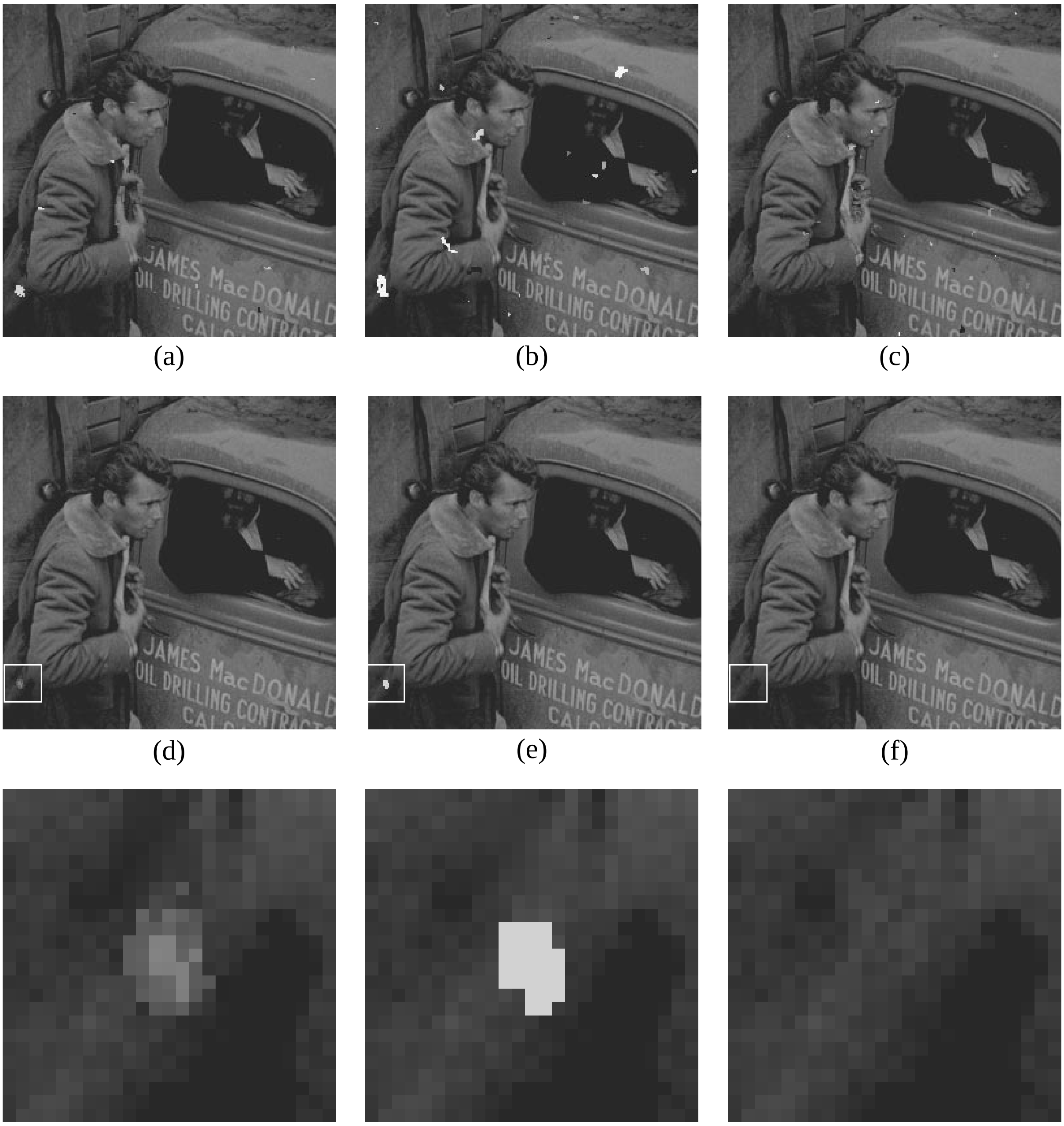
Figure 4.19 Potential function as function of iteration number: (a) for $T_{begin} = 100.0$, $T_{final} = 1.0$ and $\gamma = 0.9$, (b) for $T_{begin} = T_{final} = 1$.

$$
\begin{aligned}
& P[o \mid \boldsymbol{a}, \sigma_e^2, z_+(\boldsymbol{i}), d(\boldsymbol{i})] \\
& \propto \exp\Bigg(-\frac{1}{T} \sum_{\boldsymbol{i} \in S_r} [(1-d(\boldsymbol{i})) \cdot (z(\boldsymbol{i}) - AR(\hat{y}, \boldsymbol{a}, \boldsymbol{i}))^2 + \\
& \qquad d(\boldsymbol{i}) \cdot (o(\boldsymbol{i}) \cdot z_{mc}(\boldsymbol{i}, t+1) + (1-o(\boldsymbol{r})) \cdot z_{mc}(\boldsymbol{i}, t+1) - AR(\hat{y}, \boldsymbol{a}, \boldsymbol{i}))^2 \\
& \qquad \sum_k \beta \left| o(\boldsymbol{i}) - o(\boldsymbol{i} + \boldsymbol{q}_k) \right|] \Bigg),
\end{aligned}
\tag{4.40}
$$

Figure 4.19 shows that both experiments converge. The solution found in the full SA scheme converged to a lower potential (final potential 483) than the solution found with the Gibbs sampler only (final potential 1307). The difference is, however, that the first experiment required about 625 iterations to reach its optimum, whereas the second experiment required only 25 iterations. Visually, the corrected results are not noticeably different. The conclusion is that it is not necessary to apply an elaborate cooling schedule and that sufficiently good results can be obtained in relatively few iterations.

It must be emphasized that the result obtained by applying the Gibbs sampler only (without a cooling schedule) does not in general result in a MAP estimate. The reason why it is so successful here is probably because the distributions from which the samples are drawn are very compact; there is not a lot of ambiguity in drawing a sample.

The top row in Figure 4.20 shows three frames from the *Western* test sequence: (motion-compensated) previous, current, and next. The second row shows three corrections of the current frame, made with the *3DAR* and the *ML3Dex* methods, described in Section 4.2.3, and with the CP method described in the previous section. The results from the CP method were obtained by using just 30 iterations of the Gibbs sampler.



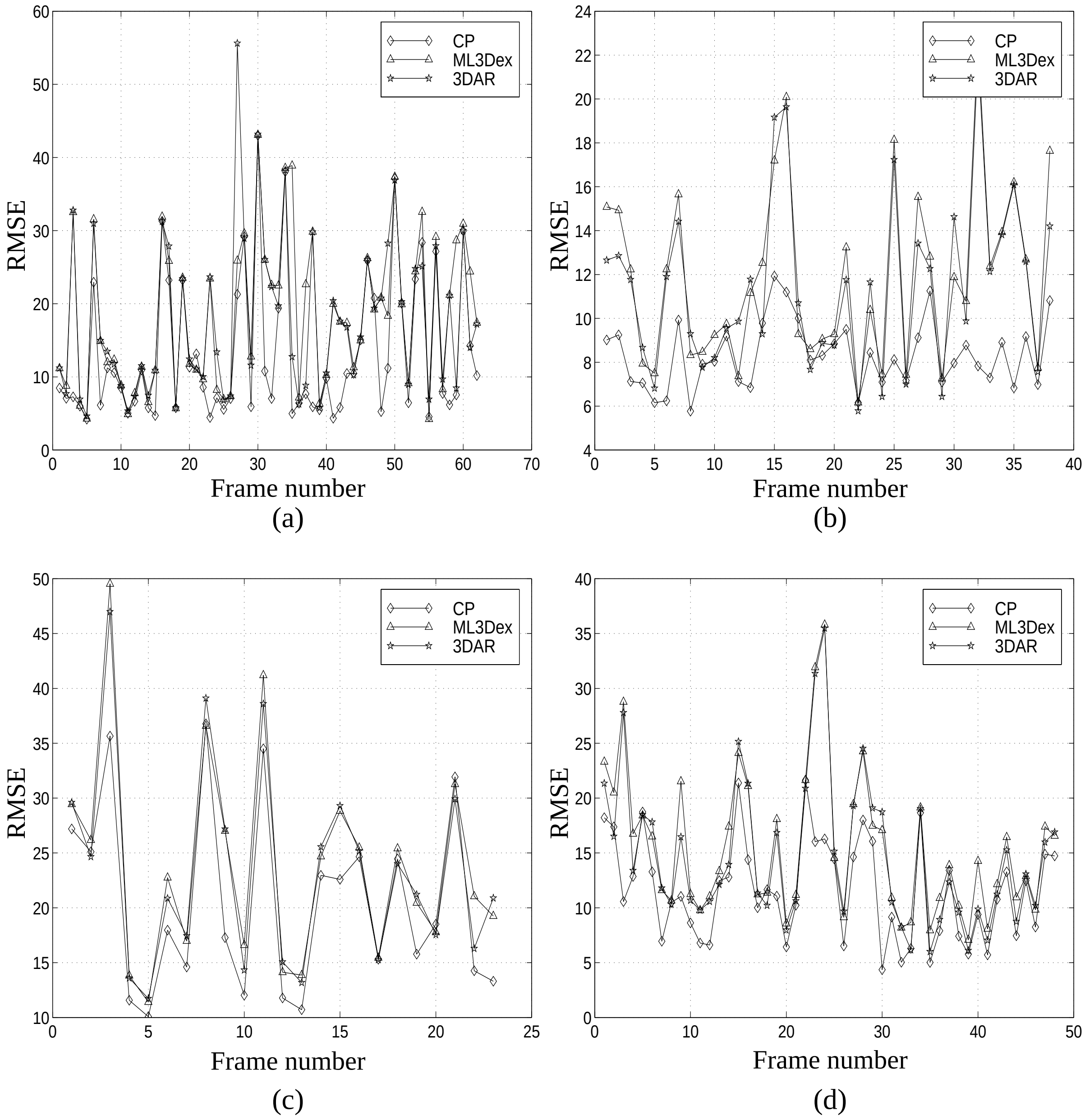
(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 4.20 (a) Motion-compensated previous frame, (b) current frame t, (c) motion-compensated next frame, (d), (e), (f) restored frame t by the 3DAR, ML3Dex, and CP schemes, respectively. Note the differences within the boxed regions. (g), (h), (i) Zoom-in to the boxed regions of panels (d), (e), and (f), respectively.

All the corrected frames show a great improvement over the corrupted frame. However, the *3DAR* and the *ML3Dex* methods fail where the motion-compensated frames are corrupted (see the highlighted boxes in the figures). These methods fail because they always incorporate data from both motion-compensated frames, regardless of the fact that some of those data may be corrupted. The *B3DAR* method, of which the results are not shown, also fails in this particular case because a block-based approach is used to determine the direction of interpolation,



(a) (b)

(c) (d)

Figure 4.21 RMSE of corrected sequences with original, unimpaired sequences: (a) Western, (b) MobCal, (c) Manege, and (d) Tunnel.

Interpolator	Western (RMSE)	Mobcal (RMSE)	Manege (RMSE)	Tunnel (RMSE)
None	113.2	81.4	86.7	90.5
ML3Dex	20.8	12.6	25.2	16.7
3DAR	20.9	12.1	24.8	15.9
CP	16.1	8.5	22.1	12.4

Table 4.4 RMSE computed between the corrected and original, unimpaired sequences.

regardless on the validity of the data within the block. Figure 4.20g-i zooms in on the boxed regions. Clearly, the proposed CP method outperforms the other methods in terms of visual quality.

4.6 Results and discussion

This section evaluates the complete chain of blotch detection, postprocessing, motion vector repair, and interpolation as depicted in Figure 4.8a. All experiments apply the SROD detector with postprocessing because this gives the highest ratio of correct detections to false alarms. The motion vector repair uses the block matching technique described in Section 4.2.2. Three interpolators are evaluated, namely, the ML3Dex, the 3DAR, and the CP method (using 30 iterations per frame).

Figure 4.21 shows the *root mean squared error* (RMSE), which is defined as the squared root out of the MSE, for the test sequences as a function of frame number. For each sequence the SROD detector with postprocessing was set to an overall correct detection rate of about 85%. The RMSE was computed only at locations at which the true blotch mask or the estimated blotch mask indicate corruptions (i.e., at locations where the original image data was altered by blotches or by interpolating false alarms). Figure 4.21 indicates that the CP interpolation method has the best performance. Whether the ML3Dex performs better than the 3DAR method is difficult to determine from this figure. Table 4.4 lists RMSE computed over all frames. It can be seen from this table that the interpolation considerably decreases the average errors. These data confirm that the CP method gives the best performance. Furthermore, it can be seen that the ML3Dex method, on average, performs slightly better than the AR method.

In terms of computational load, the CP method is to be preferred to the 3D AR method. The 3D AR method requires a matrix to be inverted, see (4.20), the size of which increases with increasing blotch size. Therefore, the number of computations for this method grows exponentially (order 3) [75,92] with increasing blotch size. There is also a risk that the system in (4.20) is singular and that no unique solution exists. In such cases, singular value decomposition [75,92] is useful. The ML3Dex interpolator is, computationally speaking, the most efficient: it is a non-iterative method that has to be evaluated only at the locations containing missing data, and it can be implemented efficiently with fast sorting algorithms [75].

The methods for blotch detection and correction introduced in this chapter give significantly better results than those obtained by existing methods. However, as can be seen from the ROCs in Figure 4.13, the ratio of false alarms to correct detection remains relatively high for some sequences. There is room for further improvements. Nonetheless, even though too many false alarms are generated in some cases, the methods described in this chapter are very useful and can be applied efficiently in practical situations. Visually disturbing artifacts introduced into a corrected sequence due to false alarms can be removed by manual intervention. Removing regions of false alarms and undoing erroneous interpolations by single mouse clicks is much more efficient than having an operator mark and correct blotches in image sequences manually.

Chapter 5

Noise reduction by coring

Summary. Coring is a well-known technique for removing noise from images. The mechanism of coring consists of transforming a signal into a frequency domain and reducing the transform coefficients by the coring function. The inverse transform of the cored coefficients gives the noise-reduced image. This chapter develops a framework for coring image sequences. The framework is based on 3D image decompositions, which allows temporal information to be exploited. This is preferable to processing each frame independently of the other frames in the image sequence. Furthermore, this chapter shows that coring can be imbedded into an MPEG encoder with relatively little additional complexity. The adjusted encoder significantly increases the quality of the coded noisy image sequences.

5.1 Introduction

As computers with memories sufficiently large to store images and even short image sequences became widespread some 25 years ago, many researchers began to investigate digital algorithms for noise reduction. The well-known theories developed by Wiener and Kalman for optimal linear filtering were applied in the digital domain on a large scale. New types of nonlinear filters, such as order statistics filters and switching filters, were developed. Nowadays many very different approaches towards noise reduction are found in the literature [1,5,13,20,22,35,46,69,70,85]. One such approach that has gained great popularity in recent years and that has proven to be very successful for denoising 2D images is *coring*. This chapter investigates this method for noise reduction and extends its application to image sequences.

Coring is a technique in which each frequency component of an observed signal is adjusted according to a certain characteristic, the so-called coring function. Originally coring was developed as a heuristic technique. It was first applied in 1951 for removing spurious oscillations in the luminance signal that were caused by a system designed to make television pictures more crisp [30]. In 1968 it was recognized that this technique could also be used for removing imperfections such as noise from signals [60]. In the 1970s and the early 1980s coring was applied in the digital domain for noise reduction [2,72,87]. The technique of *thresholding* or *coring* received a lot of attention after Donoho and Johnstone applied it successfully in the wavelet transform domain [20,22] in 1994.

Section 5.2 describes techniques for optimal filtering in a *minimum-mean-squared-error* (MMSE) sense. An example of such an optimal filter, the Wiener filter, is derived. The Wiener filter is a linear filter. If the constraint of linearity is dropped, more general nonlinear filters result. The filter characteristics of these nonlinear filters are represented by coring functions.

The domain in which coring is applied determines the effectiveness of coring for noise reduction. Section 5.3 describes two spatial signal transforms. One is a bi-orthogonal wavelet transform, and the other is a directionally sensitive subband decomposition. It is shown how to extend these 2D transforms to include the temporal dimension. The spatio-temporal decomposition provides a good basis for coring image sequences.

Noise-reduced signals are often stored or broadcast in a digital format. Section 5.4 investigates how noise can be reduced and compressed simultaneously within an MPEG2 encoder by coring the DCT coefficients. Section 5.5 concludes this chapter with a discussion.

5.2 Noise reduction techniques

5.2.1 Optimal linear filtering in the MMSE sense

Any recorded signal is affected by noise, no matter how accurate the recording equipment. In this chapter noise is modeled by a additive white gaussian source. Let $y(\boldsymbol{i})$ be an original, unimpaired frame and let the noise be $\eta(\boldsymbol{i})$. The observed frame $z(\boldsymbol{i})$ is given by:

$$z(\boldsymbol{i}) = y(\boldsymbol{i}) + \eta(\boldsymbol{i}). \tag{5.1}$$

A class of linear filters are the *finite impulse response* (FIR) filters, which are defined by:

$$\hat{y}(\boldsymbol{i}) = \sum_{k=1}^{n} h_k \cdot z(\boldsymbol{i} + \boldsymbol{q}_k). \tag{5.2}$$

Here h_k, with $k = 1, \ldots, n$, are the n filter coefficients and the $\boldsymbol{q}_k$ define the support of the filter. The optimal filtering coefficients in MMSE sense can be found by:

$$\underset{h_1, \ldots, h_k}{\arg \min} E[(y(\boldsymbol{i}) - \hat{y}(\boldsymbol{i}))^2]. \tag{5.3}$$

The filter that results is known as the Wiener filter. The Wiener filter can be implemented efficiently via the Fourier domain [53,94]. Let Fourier transform of (5.1) be given by:

$$Z(\omega) = Y(\omega) + N(\omega), \tag{5.4}$$

The estimates $\hat{Y}(\omega)$ are given by:

$$\hat{Y}(\omega) = \frac{S_{yy}(\omega)}{S_{yy}(\omega) + S_{\eta\eta}(\omega)} \cdot Z(\omega). \tag{5.5}$$

Here $S_{yy}(\omega)$ and $S_{\eta\eta}(\omega)$ indicate the *power spectral density* (PSD) functions of the unimpaired signal and the noise. From (5.5) it can be seen that each frequency component of the observed data is weighted depending on the spectral power densities of the original, unimpaired signal and noise.

5.2.2 Optimal noise reduction by nonlinear filtering: coring

The Wiener filter imposes a FIR structure onto the solution of the MMSE problem. The optimal solution to the MMSE problem that is obtained when no constraints are placed on the filter structure is often a nonlinear function. Let $\hat{Y}(\omega)$ be a general function of the observed data $Z(\omega)$. The optimal estimate $\hat{Y}(\omega)$, given a single observation $Z(\omega)$, is found with the conditional expectation [55]:

$$\begin{aligned} E[(Y(\omega) - \hat{Y}(\omega))^2] &= E[E[(Y(\omega) - \hat{Y}(\omega))^2 | Z(\omega)]] \\ &= \int_{-\infty}^{\infty} E[(Y(\omega) - \hat{Y}(\omega))^2 | Z(\omega)] \cdot P[Z(\omega)] dZ(\omega). \end{aligned} \tag{5.6}$$

The integrand in (5.6) is positive for all $Z(\omega)$; therefore, the integral is minimized by minimizing $E[(Y(\omega) - \hat{Y}(\omega))^2 | Z(\omega)]$ for each ω. This minimum is given by:

$$\hat{Y}(\omega) = E[Y(\omega) | Z(\omega)]. \tag{5.7}$$

The general solution given by (5.7) yields the smallest possible mean square error for estimating $\hat{Y}(\omega)$, given a single observation $Z(\omega)$. In general, the Wiener solution will have larger mean square errors. Further development of (5.7) gives:

$$\begin{aligned}\hat{Y}(\omega) &= E[Y(\omega)|Z(\omega)] \\ &= \int_{-\infty}^{\infty} Y(\omega) \cdot P_{Y(\omega)|Z(\omega)}[Y(\omega)|Z(\omega)]dY(\omega).\end{aligned} \tag{5.8}$$

Here $P_{A|B}[A|B]$ indicates the pdf of A, given B. If the distributions of $Y(\omega)$ and $N(\omega)$ are known, then $P_{Y(\omega)|Z(\omega)}[Y(\omega)|Z(\omega)]$ can be determined via Bayes' rule:

$$\begin{aligned}P_{Y(\omega)|Z(\omega)}[Y(\omega)|Z(\omega)] &= \frac{P_{Z(\omega)|Y(\omega)}[Z(\omega)|Y(\omega)] \cdot P_{Y(\omega)}[Y(\omega)]}{P_{Z(\omega)}[Z(\omega)]} \\ &= \frac{P_N(\omega)[Z(\omega)-Y(\omega)] \cdot P_{Y(\omega)}[Y(\omega)]}{\int_{-\infty}^{\infty} P_N(\omega)[Z(\omega)-Y(\omega)] \cdot P_{Y(\omega)}[Y(\omega)]dY(\omega)}.\end{aligned} \tag{5.9}$$

In (5.7), (5.8), and (5.9), the interpretation given to ω is that of frequency. Note that this frequency need not necessarily be obtained by applying a Fourier transform to a signal. Other transforms, such as the DCT, wavelet transforms, and subband transforms, may well be used.

Figure 5.1a shows a typical characteristic that results from (5.8). This characteristic is called a coring function. Sometimes this characteristic is also referred to as *Bayesian optimal coring* because of the relationship in (5.9) [90]. In general, coring functions leave transform coefficients with high amplitudes unaltered, and the coefficients with low amplitudes are shrunk towards zero. Intuitively speaking, this is appealing. Coefficients with high amplitudes are reliable because they are influenced relatively little by noise. These coefficients should not be altered. Coefficients with low amplitudes carry relatively little information and are easily influenced by noise. Therefore, these coefficients are unreliable, and their contribution to the observed data should be reduced.

5.2.3 Heuristic coring functions

Originally, coring was developed as a heuristic technique for removing noise. Three well-known heuristic coring functions are described here.

Soft thresholding. Soft thresholding is defined by [20,72]:

$$\hat{Y}(\omega) = \begin{cases} \operatorname{sgn}((Z(\omega)) \cdot (|Z(\omega)| - T)) & \text{if } |Z(\omega)| > T \\ 0 & \text{otherwise} \end{cases}, \tag{5.10}$$

where the $\operatorname{sgn}(Z(\omega))$ gives the sign (or phase) of $Z(\omega)$. Figure 5.1b plots this coring function.

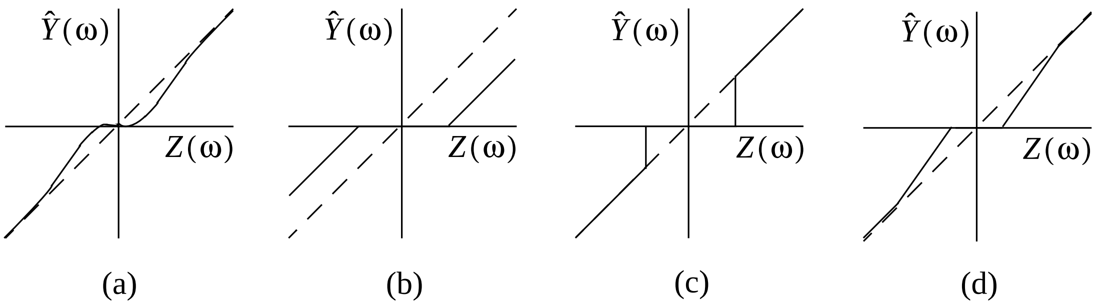


Figure 5.1 Coring functions: (a) Bayesian optimal coring, (b) soft thresholding, (c) hard thresholding, (d) piecewise linear coring.

Natural signals tend to have weak high-frequency components. Therefore, soft thresholding nullifies the high-frequency transform coefficients obtained from a signal. The result is that, besides the noise being removed, the slopes of edges are reduced and their rise time increases. For images this is perceived as blurring of edges in images. Soft thresholding has another effect, namely, it reduces contrast because it shrinks the magnitudes of all AC transform coefficients indiscriminately.

Hard thresholding. Hard thresholding is defined by [20,72]:

$$\hat{Y}(\omega) = \begin{cases} Z(\omega) & \text{if } |Z(\omega)| > T \\ 0 & \text{otherwise} \end{cases} . \tag{5.11}$$

Figure 5.1c plots this coring function. A disadvantage of hard thresholding is that it introduces spurious oscillations or so-called ringing patterns. These occur because hard thresholding not only removes noise energy at selected frequencies, but also signal energy. The removal of signal energy can be viewed as adding impulses to the original, unimpaired signal. The amplitudes of these impulses are equal to those of the original signal contents, but the signs are opposite. In the synthesis stage, where the signal is transformed back from the frequency domain to the spatial domain, the impulse responses of the synthesis filters are superimposed on the result. These superimposed filter responses are perceived as ringing.

Piecewise linear coring. A compromise between soft thresholding and hard thresholding is piecewise linear coring:

$$\hat{Y}(\omega) = \begin{cases} Z(\omega) & \text{if } |Z(\omega)| > T_1 \\ \dfrac{|Z(\omega)| - T_0}{T_1 - T_0} \cdot T_1 \cdot \operatorname{sgn}(Z(\omega)) & \text{if } T_0 \le |Z(\omega)| \le T_1 . \\ 0 & \text{Otherwise} \end{cases} \tag{5.12}$$

Figure 5.1d plots this function. Piecewise linear coring is intended to reduce the ringing artifacts resulting from hard thresholding on one hand and to preserve low-contrast picture detail, which is lost by soft thresholding, on the other hand, [72].

5.3 Coring image sequences in wavelet and subband domains

The frequency domain implementation of the Wiener filter as described in Section 5.2.1 can be viewed as an implementation of coring: each observed frequency component is adjusted according to a characteristic that is determined by the PSDs of signal and noise. However, the use of the Fourier transform as a decorrelating transform has the disadvantage of forfeiting knowledge of the spatial locations of dominant signal components. This implies that the cored signal is not adapted to local statistics, but depends on global statistics only. Clearly, this is suboptimal because local statistics can be very different from the global statistics.

The objective of transforming data prior to coring is to separate the signal from the noise as well as possible. To get optimal separation of the signal and the noise, it is advantageous to use transforms that compact the signal energy as much as possible [22,66]. Unlike the Fourier transform, scale-space representations [14, 56, 100] allow local signal characteristics at different scales to be taken into account. In the case of noise-reducing image sequences, adaptation to local statistics is advantageous due to the nonstationary, scale-dependent nature of natural images.

This section describes two 2D scale-space decompositions. The first is a nondecimated wavelet transform known as the *algorithm à trous* [36,97]. The second is a subband decomposition based on directionally sensitive filters that is known as the Simoncelli pyramid [91]. Next, it is shown how these decompositions can be extended to three dimensions by adding a temporal decomposition step. The 3D decompositions provide good separation of the signal and the noise. Which of the two scale-space-time decompositions is most suited for noise reduction by coring is investigated.

5.3.1 Nondecimated wavelet transform

The *discrete wavelet transform* (DWT) is a popular tool for obtaining scale space representations of data. A popular implementation of the DWT is the decimated DWT in which the transformed data have the same number of coefficients as the input data. A problem with this transform, however, is that shifting of the input image spatially, may lead to entirely different distributions of the signal energy over the transform coefficients [91,97]. This is caused by the critical subsampling applied in decimated wavelet transforms. Therefore, shifting the input image can lead to significantly different filtered results. This is undesirable because it can lead to temporal artifacts when in the processing of image sequences.

Shift invariance is obtained by nondecimated DWTs. An algorithm that generates nondecimated DWTs is the algorithm à trous (“algorithm with holes”) [36]. Because no subsampling is applied in this scheme, the decomposition is significantly overcomplete. For example, a three-level decomposition of an image with N pixels gives $10N$ transform coefficients.

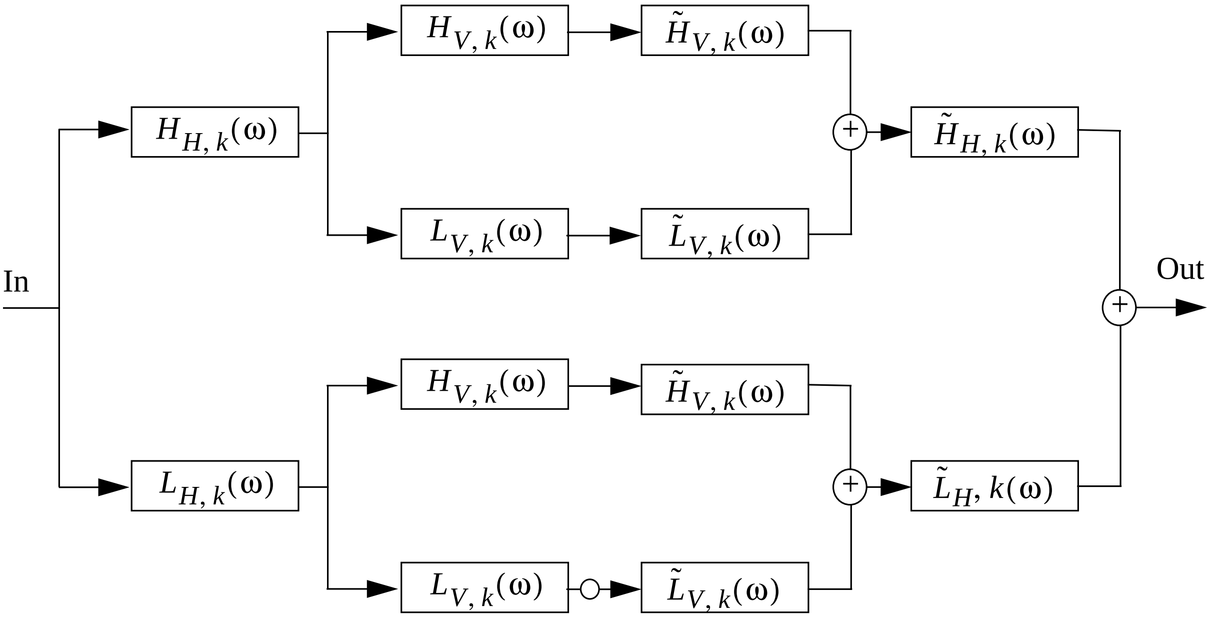

In
$H_{H,k}(\omega)$
$L_{H,k}(\omega)$
$H_{V,k}(\omega)$
$L_{V,k}(\omega)$
$H_{V,k}(\omega)$
$L_{V,k}(\omega)$
$\tilde{H}_{V,k}(\omega)$
$\tilde{L}_{V,k}(\omega)$
$\tilde{H}_{V,k}(\omega)$
$\tilde{L}_{V,k}(\omega)$
+
+
$\tilde{H}_{H,k}(\omega)$
$\tilde{L}_{H},k(\omega)$
+
Out

Figure 5.2 Overview of the algorithm à trous: a 2D wavelet analysis/synthesis scheme. The total decomposition is obtained by inserting the complete filter bank into the white spot near the bottom of the figure recursively. At each recursion level, index k is incremented.

Figure 5.2 gives a schematic overview of the algorithm à trous. First the input image is filtered twice in horizontal direction; once with the high-pass analysis filter and once with the low-pass analysis filter. Next, the filtered data are filtered again with the same high-pass and low-pass analysis filters, but now in a vertical direction. The data that result from low-pass analysis in both the horizontal and vertical directions are decomposed again with the same analysis filter banks. However, this time the analysis filters are dilated by inserting 2^{k-1} zeros between each of the filter coefficients at recursion level k, with $k = 1, 2, \ldots$ for the recursion levels. Initially, the algorithm starts out with $k = 0$, and no zeros are inserted between the filter coefficients. For the synthesis part of the filter bank, again 2^{k-1} zeros are inserted between each of the coefficients of the high-pass and low-pass synthesis filters at each recursion level k.

The algorithm à trous uses bi-orthogonal wavelet pairs. This means that synthesis filters used in the reconstruction phase are not identical to the analysis filters. Table 5.1 gives the filter coefficients for the analysis and synthesis filters. These are symmetric FIR filters, therefore they are linear phase filters. This is a useful property in image processing because nonlinear phase filters degrade edges [4].

Figure 5.3 gives the transform coefficients of the 2D algorithm-a-trous image decomposition of a test image. One half of this image consists of a frequency sweep, the other half shows half a disc that is partially contaminated by additive white gaussian noise. Figure 5.3 shows a number of things. First of all, the signal energy is concentrated in different “frequency” bands, depending on the orientation and the frequency of the local signal components. Furthermore, the spatial location of signal components is preserved; the spatial location of various signal components are clearly visible in Figure 5.3. This is in contrast to the Fourier transform, which indicates the presence of specific frequencies within a signal, but their localization is not known. Finally, the noise energy is spread out over all frequency bands and orientations.

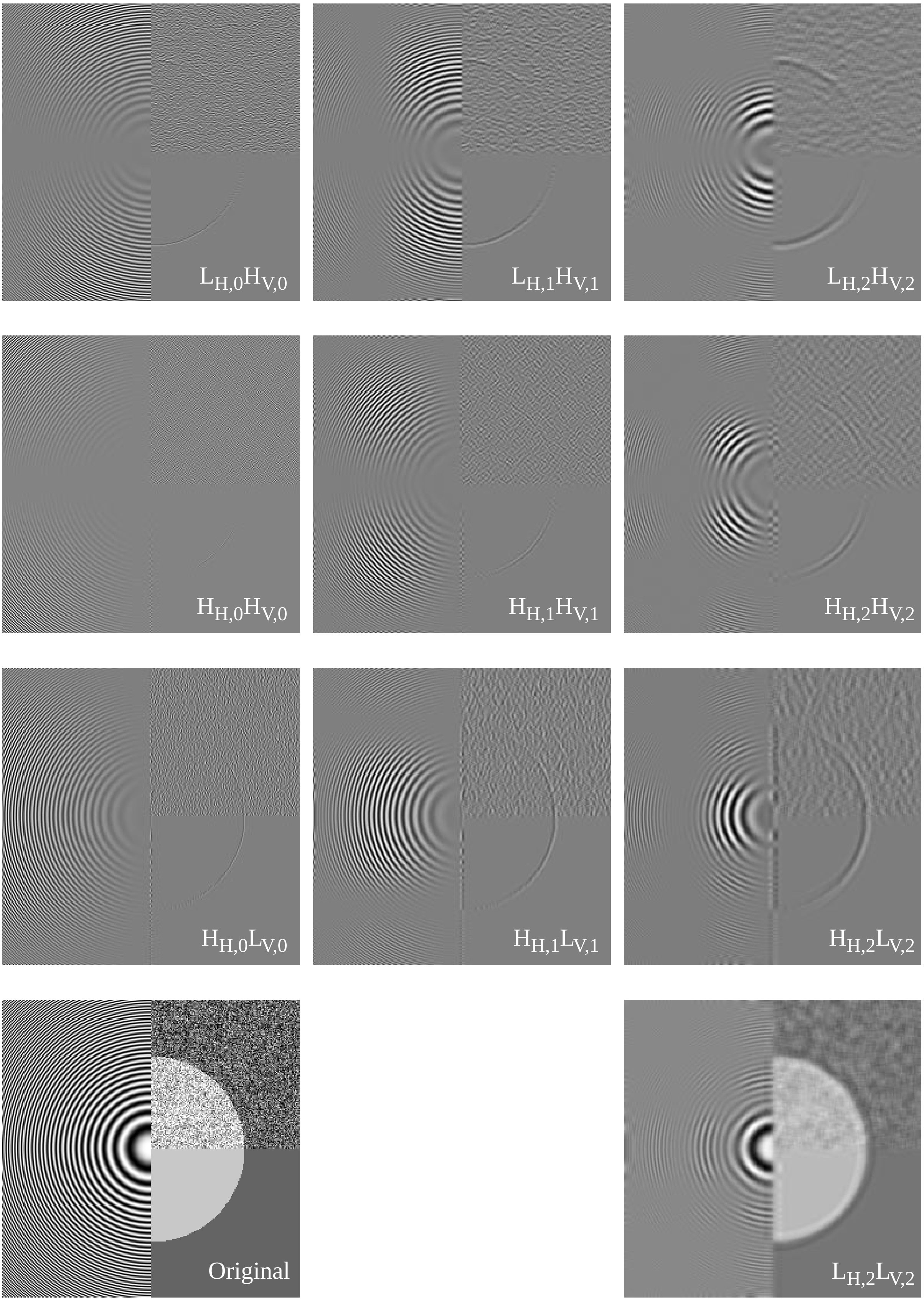


Figure 5.3 Top three rows: transform coefficients from decomposed image at levels 0, 1, and 2. Bottom right: low-pass residual. Bottom left: original input image. To improve visibility, the contrast has been stretched for all images.

Low-Pass Analysis	-1/8	2/8	6/8	2/8	-1/8
Low-Pass Synthesis		1/4	1/2	1/4	
High-Pass Analysis		1/8	-2/8	1/8	
High-Pass Synthesis	1/4	2/4	-6/4	2/4	1/4

Table 5.1 Coefficients for the bi-orthogonal wavelet pairs used by the algorithm à trous.

The filter banks used by the algorithm à trous are quite short and they are therefore not ideal in terms of cut-off frequency and signal suppression in the stop bands. The result is *spectral leakage***.** Figure 5.3 shows that energy from high-frequency signal components are visible in low-pass subbands and vice versa.

5.3.2 Simoncelli pyramid

The Simoncelli pyramid is a subband decomposition scheme based on directionally sensitive filters [91]. This means that the distribution of signal energy over frequency bands depends on the orientation of structures within the image. Shift invariance is accomplished by avoiding aliasing effects by ensuring that no components with frequencies larger than $\pi/2$ are present before 2:1 subsampling. The Simoncelli decomposition is significantly overcomplete; the number of transform coefficients is much larger than the number of pixels in the original image. For example, a four-level pyramid decomposition with four orientations (four times four sets of high-pass coefficients and one set of low-pass coefficients) of an image with N pixels gives about $9.3N$ coefficients.

Figure 5.4 shows the 2D Simoncelli pyramid (de)composition scheme. The filters $L_k(\omega)$, $H_k(\omega)$, and $F_m(\omega)$ are the 2D low-pass, high-pass, and directional (fan) filters, respectively. The filters $L_0(\omega)$, $H_0(\omega)$, $L_1(\omega)$, and $H_1(\omega)$ are self-inverting, linear-phase filters. Self inverting-filters have the pleasant property that the analysis and the corresponding synthesis filters are identical.

The following constraints apply to $L_0(\omega)$, $H_0(\omega)$, $L_1(\omega)$, and $H_1(\omega)$: the aliasing in the low-frequency (subsampled) bands is minimized (5.13), all radial bands have a bandwidth of one octave (5.14), and the overall system has unity response, requiring that low and high-pass filters are power complementary (5.15):

$$L_1(\omega) \to 0 \qquad \text{for } \omega > \frac{\pi}{2}, \tag{5.13}$$

$$L_0(\omega) = L_1(2\omega), \tag{5.14}$$

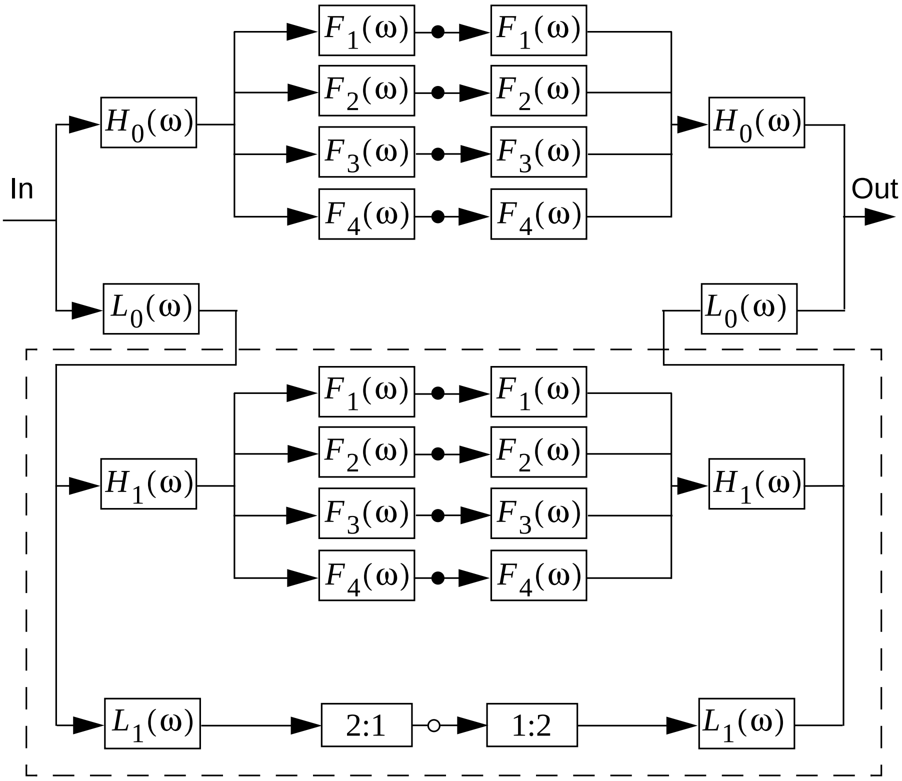


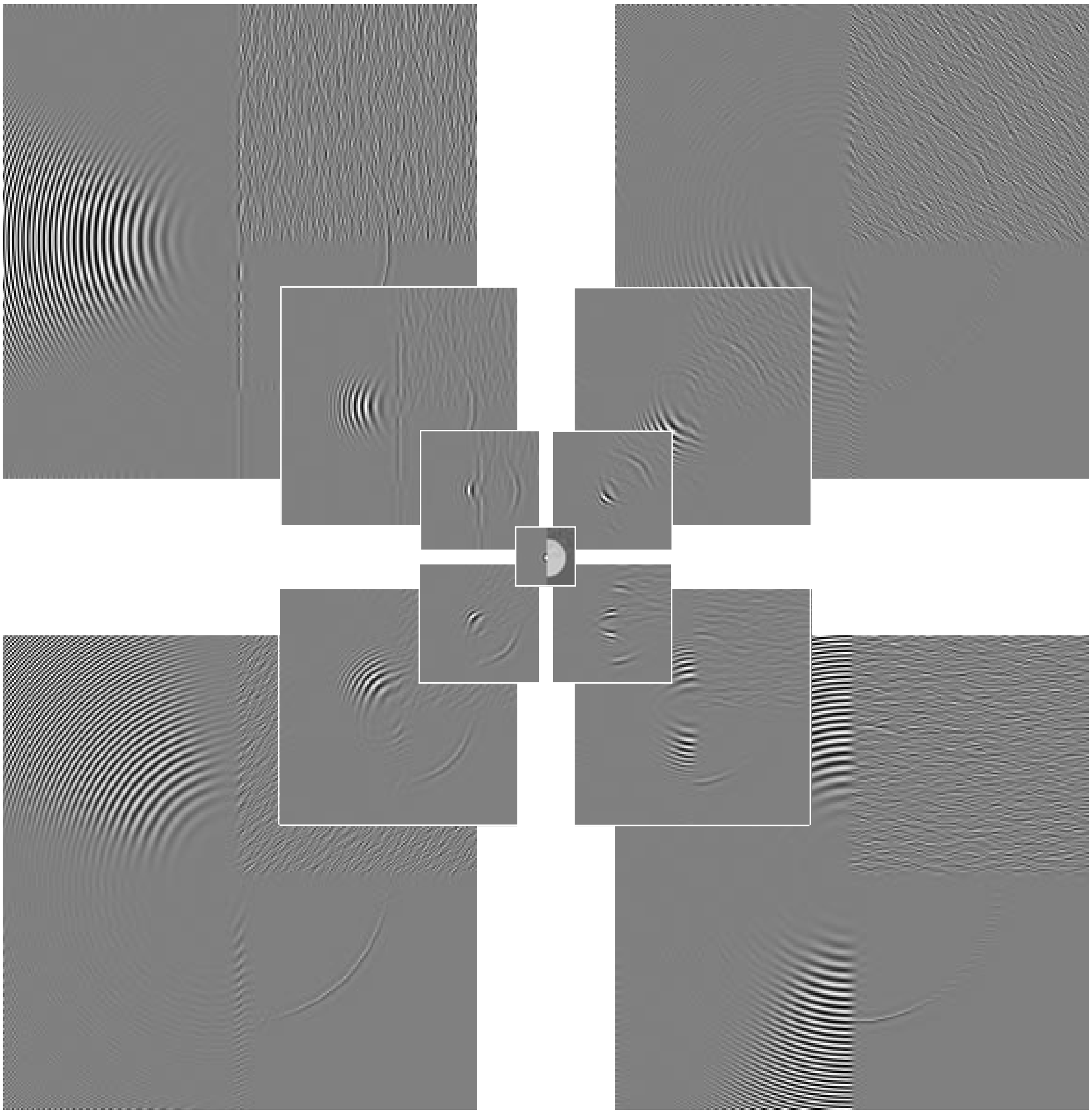
Figure 5.4 The Simoncelli analysis/synthesis filter bank. The total decomposition is obtained by recursively inserting the contents of the dashed box into the white spot near the bottom of the figure.

$$|L_i(\omega)|^2 + |H_i(\omega)|^2 = 1\,. \tag{5.15}$$

The 2D filters can be obtained from 1D linear phase FIR filters by means of the McCLellan transform [59]. Equation (5.14) can be used to obtain the 2D filter $L_0(\omega)$ from $L_1(\omega)$. A conjugate gradient algorithm was used to find the filters $H_0(\omega)$ and $H_1(\omega)$ under the constraints set by (5.15) [75].

For practical purposes, the high-pass filters $H_o(\omega)$ and $H_1(\omega)$ are directly combined with the fan filters $F_1(\omega), F_2(\omega), F_3(\omega)$, and $F_4(\omega)$. Taking the 2D Fourier transforms of H_o and H_1, multiplying the transform coefficients with $f(\theta - \theta_m)$ in (5.16), and taking the inverse Fourier transforms gives the required combination. In (5.16), θ_m is the center of the orientation of the filter.

$$f(\theta - \theta_m) = \begin{cases} 1 & \forall \; |\theta - \theta_m| < \frac{\pi}{16} \\ \cos(4 \cdot |\theta - \theta_m|) & \forall \; \frac{\pi}{16} \le |\theta - \theta_m| < \frac{3\pi}{16} \\ 0 & \text{otherwise} \end{cases} \tag{5.16}$$



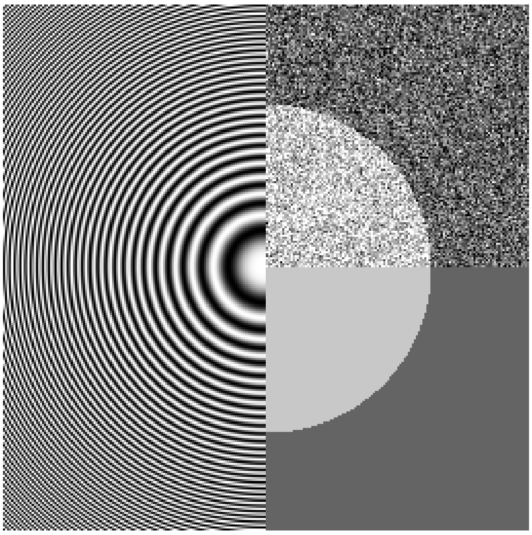


Figure 5.5 Top: pyramid decomposition of the input image showing the output of the directionally sensitive fan filters and the residual low-pass image. The contrast has been stretched to improve visibility. Note that the local signal energy is concentrated in one or two orientations, whereas the noise energy is spread out over all orientations. Bottom: test image that was also used in Figure 5.3.

The first filtering stage with filters $L_0(\omega)$ and $H_0(\omega)$ is omitted for the experiments in this chapter to reduce the number of computations. This also reduces the number of transform coefficients by $4N$, where N is the number of pixels in a frame. Figure 5.5 shows an example of a decomposition using a pyramid with three levels and the same test image as in the previous section. The 2D filter used banks consisting of 21×21 taps.

The 2D pyramid decomposition with four orientations has a number of advantages over a 2D nondecimated DWT. First, the local separation between signal and noise is better for the pyramid decomposition than for the DWT. At each level of the pyramid decomposition, the noise energy is distributed over four frequency bands, and the energy of the image structures, such as straight lines, is distributed over one or two frequency bands. In contrast, at each decomposition level of the DWT, the energy of the image structures is distributed over two or three frequency bands, and the noisy energy is also distributed over three frequency bands. Exceptions are horizontal and vertical image structures; their energy is concentrated in one frequency band only. Improved separation between signal and noise means removing more noise and distorting the signal less.

The second advantage of the 2D pyramid decomposition is that the three-level pyramid decomposition gives $5.3N$ coefficients; this is less overcomplete than a shift invariant nondecimated DWT that gives $10N$ coefficients for the same number of levels.

Finally, for the particular implementation of the Simoncelli pyramid in this chapter, there is much less leakage than for the algorithm à trous. This is a result of the constraint set by (5.13) in combination with the relatively large filter banks.

5.3.3 An extension to three dimensions using wavelets

The 2D decorrelating transforms described in the previous sections spatially separate signals from noise. It will be made apparent that this separation can be improved by including motion-compensated temporal information. If the signals are stationary in a temporal direction, the motion-compensated frames from $t - n, \ldots, t + m$ should all be identical to frame t, except for the noise terms. The (linear) pyramid decompositions of these images should also be identical, again except for the noise terms. This means that a set of transform coefficients at scale-space locations corresponding in a temporal sense should consist of a 1D DC signal plus noise. This signal can be separated into low-pass and a high-pass terms, e.g., by the DWT.

Note that, ideally speaking, one would use long analysis filters to obtain good separation of the signal and noise components in the temporal decomposition step. However, inaccuracies of the motion estimator and the fact that areas become occluded or uncovered form a limiting factor to the length of temporal filter used.

The steps to a 3D spatial-temporal decomposition/reconstruction scheme are summarized by Figure 5.6. Large reductions in computational effort can be obtained for steps 4a and 5a by realizing that, for the purposes of this chapter, it is only necessary to reconstruct the current frame t. Reconstructions of decomposed motion-estimated frames are not of interest here, and therefore they need not be computed.

Analysis:

1a. Calculate the motion compensated frames from frames at $t - n, \ldots, t + m$.
2a. Calculate a 2D decomposition for each (motion compensated) frame.
3a. Apply the DWT in the temporal direction to each set of coefficients at corresponding scale-space locations.

Synthesis:

4a. Apply the inverse DWT in the temporal direction to each set of wavelet coefficients to reconstruct the coefficients of the spatially decomposed frame at t.
5a. Apply the synthesis stage of the 2D filter bank.

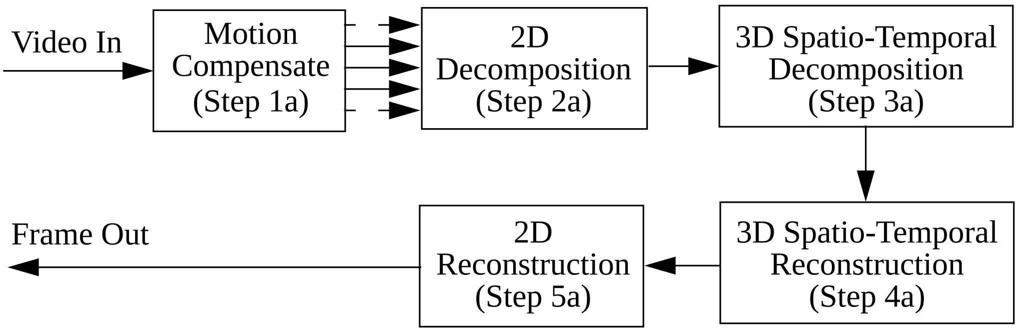

Video In
Motion Compensate (Step 1a)
2D Decomposition (Step 2a)
3D Spatio-Temporal Decomposition (Step 3a)
Frame Out
2D Reconstruction (Step 5a)
3D Spatio-Temporal Reconstruction (Step 4a)

Figure 5.6 Summary of 3D signal decomposition scheme.

5.3.4 Noise reduction by coring

The structure of the proposed decomposition/reconstruction algorithm offers several possibilities for coring transform coefficients by inserting the steps summarized in Figure 5.7. This figure represents a framework for 3D scale-space noise reduction by coring.

The generally nonlinear nature of coring makes it difficult to determine the combination of coring characteristics for steps 2b, 3b, and 4b required for optimal noise reduction. Another question is whether optimal coring requires coring in all steps 2b, 3b, and 4b. To exploit the temporal decomposition, coring is certainly necessary in step 3b. However, this alone cannot be optimal as is explained in a moment. The conclusion is that spatial noise reduction is required as well.

Coring the spatio-temporal transform coefficients only (step 3b) is suboptimal because this coring operation can be viewed as a *switching filter* [46] that turns itself on and off automatically, depending on the accuracy of the data. Suppose coring is applied to the spatio-temporally decomposed signal, and suppose there is an error in the motion estimation process that results in large frame differences. In such a case, all the coefficients resulting from the spatio-temporal decomposition have high amplitudes. The spatio-temporal coring function tends to keep high amplitudes intact and will not remove a lot of noise in such circumstances; the filter is effectively switched off locally.

Coring:

2b. Core the spatial transform coefficients (except those in the DC band) for all frames.
3b. Core the high-pass spatio-temporal transform coefficients.
4b. Core the spatial transform coefficients (except those in the DC band) of the current frame.

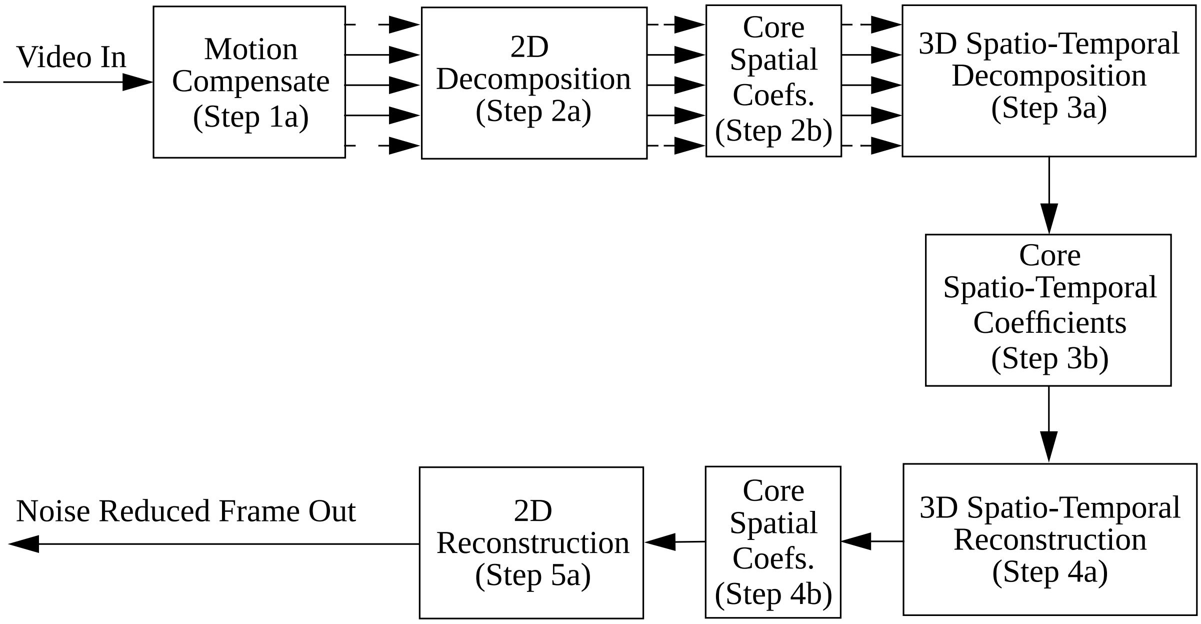

Video In
Motion Compensate (Step 1a)
2D Decomposition (Step 2a)
Core Spatial Coefs. (Step 2b)
3D Spatio-Temporal Decomposition (Step 3a)
Core Spatio-Temporal Coefficients (Step 3b)
3D Spatio-Temporal Reconstruction (Step 4a)
Core Spatial Coefs. (Step 4b)
2D Reconstruction (Step 5a)
Noise Reduced Frame Out

Figure 5.7 Representation of the 3D scale-space system for noise reduction.

Step 2b applies optimal coring functions that are computed by (5.8) for each subband of a 2D decomposition. This requires estimating or assuming distributions for the signal and noise coefficients in each subband. In step 3b, hard thresholding is used for coring the spatio-temporal coefficients because it fits in nicely with the switching filter idea. If a spatio-temporally decomposed coefficient is small, it is likely to be noise and it should be removed completely. If the coefficient is large, it is likely that the data were not stationary in a temporal sense and the coefficient should not be altered.

The optimal coring functions in step 4b are much harder to determine because they depend on the spatial coring applied in step 2b and on the spatio-temporal coring applied in step 3b. The effect of the latter is particularly difficult to model due to its dependency on the quality of motion-compensated images. Therefore, rather arbitrarily, soft thresholding is applied in step 4b. Note that soft thresholding is preferred over hard thresholding because the latter tends to give disturbing ringing patterns as discussed earlier.

Threshold selection. A good value for the hard thresholding in the spatio-temporal threshold is $T_{st} = 3 \cdot \sigma_{hp}$, where σ_{hp}^2 is the estimated variance of the noise in the high-pass coefficients. The motivation for this is the following. If the noise corrupting the image sequence is assumed to be additive, white, and gaussian, and if the motion compensation is perfect, then the high-pass coefficients contain noise energy only. In fact, the high-pass coefficients follow a zero-mean gaussian distribution. Setting all observed coefficients that lie within $\pm 3 \cdot \sigma_{hp}$ to zero effectively means that noise is removed from 99.7% of the high-pass coefficients. It is assumed that the variance of the high-pass coefficients is much greater than that of the noise, if the motion compensation is not perfect. Therefore, if the temporal intensity differences are large due to imperfect motion compensation, the signal will hardly be affected by the temporal coring.

The threshold T_s for soft thresholding the spatial decomposition coefficients in step 4b is chosen so that the PSNR of the corrected sequence is maximal. In practical situations, T_s cannot be chosen to give the maximum PSNR due to the absence of an unimpaired original to serve as a reference. In this case, the value for T_s that gives the best visual quality of the noise reduced sequence is selected.

Note that no threshold selection is required for the coring of spatial decomposition coefficients in step 2b because the coring functions are completely determined by the signal and noise distributions in each subband.

5.3.5 Perfect reconstruction?

One of the characteristics of wavelets is that they allow for perfect reconstruction. Hence, the algorithm à trous gives perfect reconstruction. Unfortunately, this is not the case for the Simoncelli pyramid. The Simoncelli pyramid is a linear phase function, self-inverting and power complementary in the ideal case. In practice, self-inverting linear-phase FIR filters with more than two taps cannot possess both the power-complementary property and the *perfect reconstruction* (PR) property [95]. If the power-complementary property is retained, the absence of PR is reflected by ringing near sharp edges in reconstructed images. The errors introduced due to the lack of PR is represented by the difference between the original and reconstructed images.

The following investigates how the effects of lack of PR can be minimized. Let Z and $\hat{Y}$ denote a decomposed noisy image before and after coring, respectively. If the (linear) reconstruction operator is denoted by $R[\cdot]$, the noise reduced image $\hat{y}$ is given by:

$$\begin{aligned} \hat{y} &= R[\hat{Y}] \\ &= R[Z + \hat{Y} - Z] \\ &= R[Z - \hat{N}(Z)] \\ &= R[Z] - R[\hat{N}(Z)], \end{aligned} \tag{5.17}$$

where $\hat{N}(Z) = Z - \hat{Y}$ can be regarded as an estimate of the noise realization that corrupts the original data. Ideally speaking, $R[Z]$ equals z, therefore:

$$\hat{y} = z - R[\hat{N}(Z)] \tag{5.18}$$

This result shows that reconstructing an image of the noise realization and subtracting it from the noisy input image reduces the effects of lack of PR. This is done instead of directly reconstructing the noise-reduced image from the cored transform coefficients. Hence, the problem

Sequence	Noisy Sequence (dB)	à trous Spatial Coring (dB)	à trous Temporal+Spatial Coring (dB)	Pyramid Spatial Coring (dB)	Pyramid Temporal+Spatial Coring (dB)
Plane 25	33.0	3.6	4.1	3.8	4.8
Plane 100	27.0	4.8	6.1	5.7	6.7
Plane 225	23.5	6.0	8.3	7.0	8.5
MobCal 25	33.0	1.8	2.6	1.6	2.9
MobCal 100	27.0	3.4	4.7	3.5	4.9
MobCal 225	23.5	4.6	5.9	4.9	6.1

Table 5.2 PSNR of test sequences and increase in PSNR of noise reduced sequences using the Pyramid and Wavelet decomposition schemes with and without coring of spatio-temporal subband coefficients.

of lack of PR for the original image is shifted to lack of PR for the noise realization. This approach, however, introduces no artifacts, such as ringing, that are associated with image structures if the noise is independent of the image contents. Furthermore, because the noise has a lower variance than that of the image contents, the effects of lack of PR for the reconstructed noise signal are much less (or not) visible.

5.3.6 Experiments and results

This section evaluates the noise-reduction capabilities of the wavelet and pyramid noise-reduction schemes described in Section 5.3.4. In both cases, the 2D decompositions are extended to three dimensions by the same bi-orthogonal wavelet used by the algorithm à trous (Table 5.1). To get some indication of the gains achieved by 3D filtering over 2D filtering, the test sequences are processed twice by each filter: once with coring of the spatio-temporal decomposed coefficients (step 3b) and once without. To reduce the computational complexity, no coring is applied to the spatially decomposed motion-compensated frames, i.e., step 2b is omitted.

Two test sequences are evaluated in this section. The first sequence is called *Plane* and shows a plane flying over a landscape. It contains fine detail, sharp edges, uniform regions, and a lot of motion. The sequence was originally recorded with a high-definition camera, and the images are very crisp. There are strong interlacing effects due to motion. The second test sequence is the well-known *MobCal* sequence, which does not display noticeable interlacing effects. Ideally, to avoid the effects of interlacing, one would apply motion-compensated de-interlacing [19]. The noise-reduction filters would be applied to the de-interlaced frames. However, motion-compensated de-interlacing adds a lot of complexity to the noise-reduction system. Therefore, the *Plane* sequence is processed on a field-by-field basis instead of on a frame-by-frame basis.

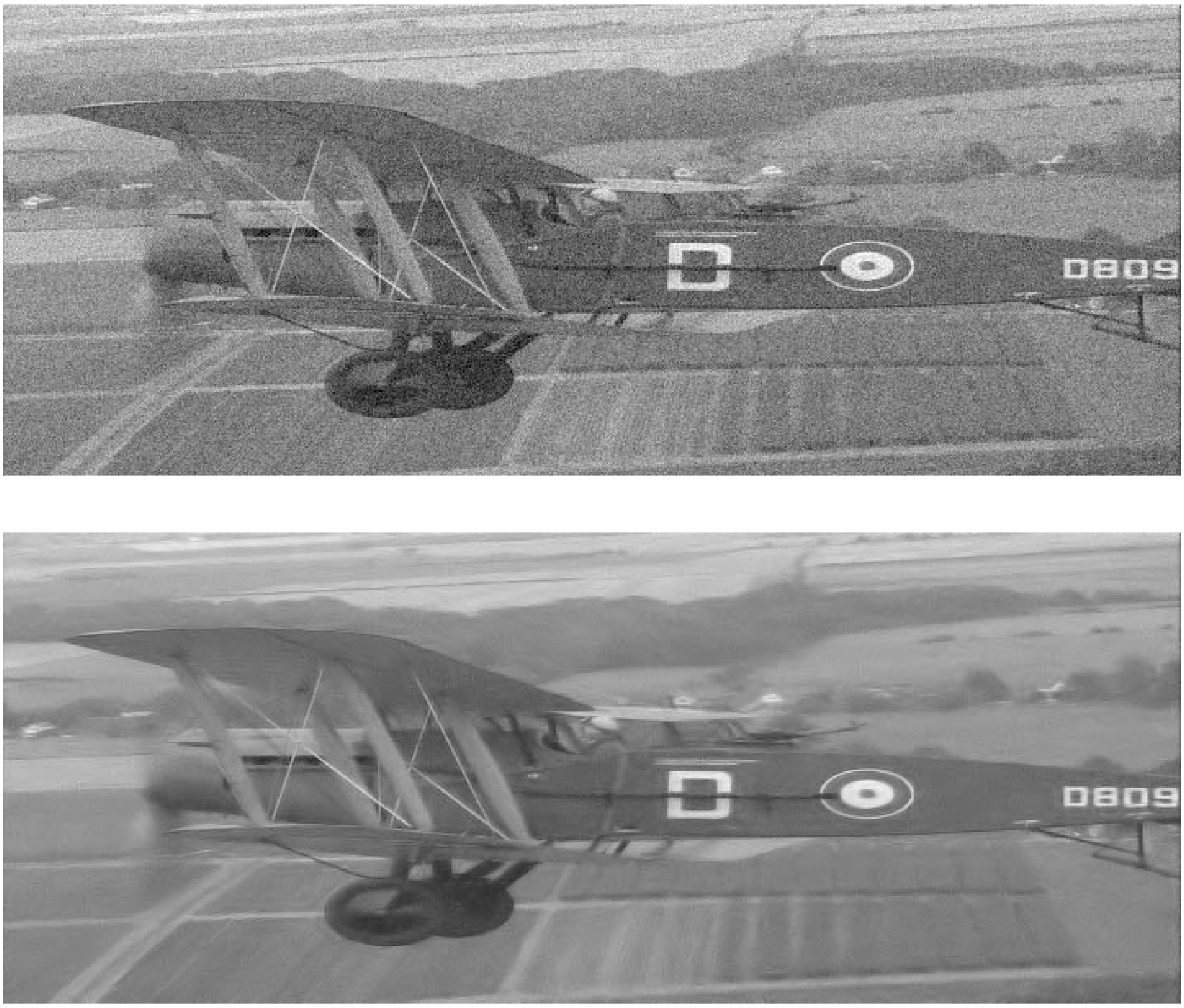


Figure 5.8 Top: noisy field from *Plane* sequence with noise variance 225 (PSNR = 23.5 dB). Bottom: filtered result from the 3D pyramid filter. (Sequence available by courtesy of the BBC).

White gaussian noise, with variances 25, 100, and 225, has been added to the test sequences. Figure 5.8 shows an example of a noisy field from the *Plane* sequence and the filtered result obtained by 3D pyramid with both spatio-temporal and spatial coring. Table 5.2 lists the PSNRs of the test sequences and the increase in PSNR for the filtered results.

Considerable amounts of noise reduction are achieved by the filters. The best results are obtained by coring both the spatio-temporal coefficients and the spatial coefficients (step 3b and step 4b in Section 5.3.4), which gives an improvement ranging from 0.5 to 2.3 decibels over spatial filtering only. The magnitude of the improvements depend on the sequence, the amount of noise, and the spatio-temporal decomposition used. The performance of the pyramid filter is similar or better than that of the shift invariant wavelet filter in terms of PSNR in all cases. Visually speaking, the results given by the pyramid filter are better than those of the wavelet filter; the results are a bit sharper, and artifacts that result from filtering in the form of “low-frequency spatial patterns” are less visible.

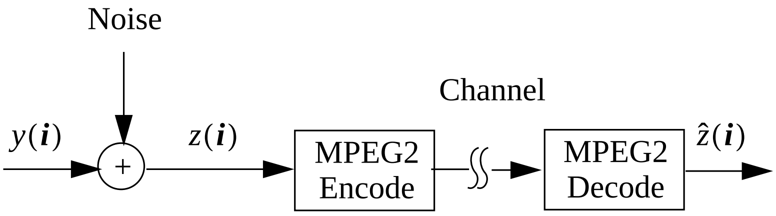


Figure 5.9 MPEG2 encoding of noisy image sequences.

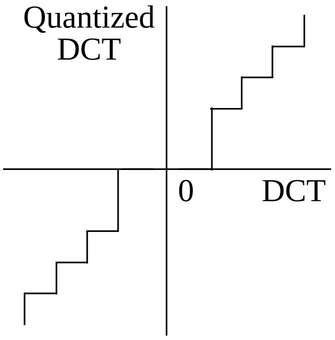


Figure 5.10 Example of quantization of AC DCT coefficients with a dead-zone around zero.

5.4 MPEG2 for noise reduction

Consider a broadcasting environment in which noisy film and video sequences are digitally broadcast with an MPEG2 encoding system, as illustrated by Figure 5.9. It is assumed that no channel errors are introduced. MPEG2 encoding systems try to minimize the coding errors between input $z(\boldsymbol{i})$ and output $\hat{z}(\boldsymbol{i})$. However, in the case of noisy image sequences, what they should be doing is minimizing the errors between the original, noise-free image $y(\boldsymbol{i})$ and the output $\hat{z}(\boldsymbol{i})$. When doing so, the MPEG2 encoding systems can be considered devices for simultaneous noise reduction and image compression.

Let $\varepsilon(\boldsymbol{i})$ denote the error between $y(\boldsymbol{i})$ and $\hat{z}(\boldsymbol{i})$. The aim of this section is to adjust an MPEG2 encoding system to minimize the error variance. The error variance can be expressed in terms of DCT coefficients:

$$\begin{aligned} E[\varepsilon^2(\boldsymbol{i})] &= E[(y(\boldsymbol{i})-\hat{z}(\boldsymbol{i}))^2] \\ &= \sum_{k=1}^{64} E[(Y_k(\boldsymbol{i}')-\hat{Z}_k(\boldsymbol{i}'))^2]. \end{aligned} \tag{5.19}$$

Here $Y_k(\boldsymbol{i}')$ and $\hat{Z}_k(\boldsymbol{i}')$, with $k = 1, \ldots, 64$, represent the 64 DCT coefficients of each 8×8 data block within a frame. The column, row, and frame number of a data block is indicated by $\boldsymbol{i}'$.

Two basic approaches can be followed to minimize (5.19). In theory, these approaches give the same results. The first approach directly minimizes $E[(y(\boldsymbol{i})-\hat{z}(\boldsymbol{i}))^2]$. As is shown in Appendix C, this approach is equivalent to determining optimal quantizers for the DCT coefficients of a noisy signal. The second approach is based on the fact that the problem of minimizing the overall error variance can be split into two parts for a communication system in which a signal is distorted prior to (lossy) channel encoding [103]. The first part consists of computing the conditional expectation for the true signal given the observed noisy data. The second part consists of designing an encoder that is optimal for the original, noise-free signal.

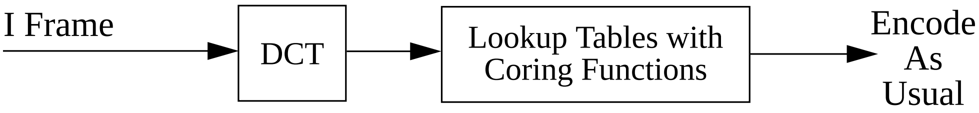


Figure 5.11 Coring of the DCT coefficients of I frames in an MPEG2 encoder.

In the particular case of Figure 5.9, the advantage of the second approach is that, in principle, the encoder is already optimized for encoding noise-free signals. Therefore, it is not necessary to design new quantization tables as is required for the first approach. All that needs to be done for the second approach is to core the DCT coefficients following (5.8) prior to quantization, i.e., by replacing the observed DCT coefficients with the conditional expectation for the true DCT coefficients. This second approach is investigated further in this section.

In fact, MPEG2 encoders implicitly core noisy DCT coefficients to some extent by incorporating a so-called *dead zone* in the quantizers for the coefficients of the non-intra-coded frames (Figure 5.10) [61]. As a result of the dead zone, DCT coefficients with small magnitudes are mapped to zero. However, note that the use of dead zones is suboptimal for noise reduction because they are not applied to all frames and because they do not address the noise on DCT coefficients with larger amplitudes.

5.4.1 Coring I, P, and B frames

I-frames. The MPEG2 system defines three frame types; namely, I frames and predicted P and B frames. The I frames are encoded by dividing the frames in 8×8 blocks, applying the DCT to the blocks and quantizing the DCT coefficients (Chapter 2). Two basic approaches can be followed towards coring the DCT coefficients of I frames. The first is to estimate the pdf for each DCT coefficient from the observed data for each frame, compute the conditional expectation for each coefficient according to (5.8), and replace the observed coefficients by these values. Computing optimal coring functions for each I frame of an image sequence is expensive in terms of computational complexity, and therefore it is expensive to implement in real-time hardware.

The second approach does not optimize the coring functions for each frame. Instead, fixed sets of coring functions are computed off-line and stored in the encoder as lookup tables (Figure 5.11). The coring functions are computed from a large set of images, so that on average the encoder gives the best results that can possibly be achieved under the condition of static lookup tables. This approach can be implemented in an MPEG2 encoder easily. Section 5.4.2 gives the details of this second approach.

B and P frames. The B and P frames are predicted from frames coded previously. The frame differences between the predicted and current frames are encoded like I frames, i.e., by using DCTs and quantization. Finding the ideal coring coefficients is more difficult now because the signal and noise distributions of the frame differences are not known. These depend on the nonlinear coring and quantization of the frames coded earlier and on the quality of the motion estimation and compensation.

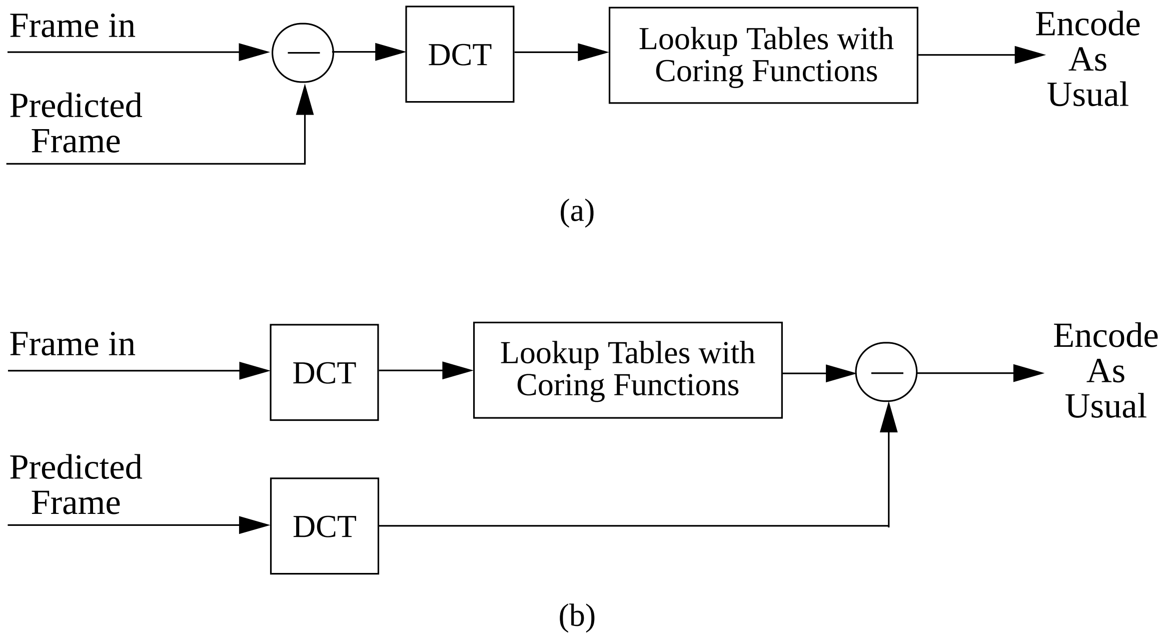


Figure 5.12 (a) Coring function applied to DCT of frame differences, (b) illustration of how, by sliding the DCT and the coring function in front of the subtraction, B and P frames can be cored as the I frames are. Note that the predicted frame is extracted from a coded frame that has already been noise reduced and need not be cored again.

Instead of coring the DCTs of the frame differences, as illustrated in Figure 5.12a, an alternative strategy is preferred in which the DCT and coring operation are performed prior to subtracting the current and predicted frames from each other. Figure 5.12b illustrates this alternative strategy. Note that the coring functions in Figure 5.12a,b are different from each other and that also the results given by the two approaches generally speaking are not identical.

Two points about the scheme in Figure 5.12b are noteworthy. First, the predicted frames have already been coded and hence they have already been noise reduced earlier on. Therefore it is not necessary to core the predicted frames again. Second, the optimal coring characteristics are identical to those computed before for the I frames. This means that only one set of lookup tables is required for the I, P and B frames.

5.4.2 Determining the DCT coring functions

This sections deals with computing the coring functions for the I, P, and B frames. As indicated in the previous section, the coring functions are computed from a large set of images, so that the encoder gives the best results that can be achieved on average with static lookup tables. Computing the coring functions consists of two steps. First, the distributions of the signal and the noise have to be determined. Next, the coring functions can be computed from (5.8) and (5.9).

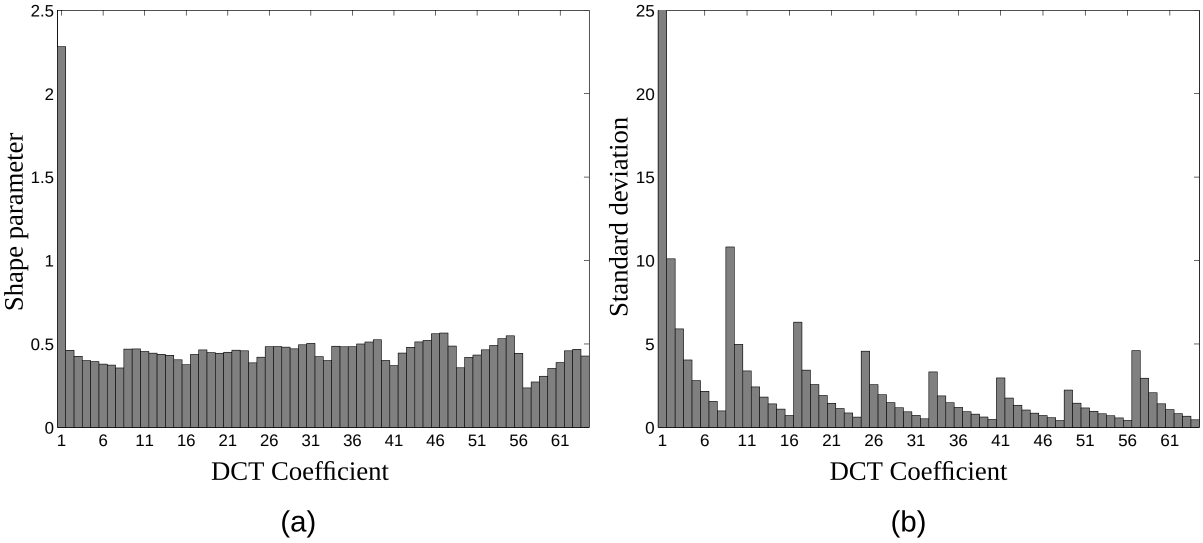


Figure 5.13 (a) Shape parameters and (b) standard deviations estimated for the DCT coefficient.

The noise corrupting the image sequences is assumed to be additive, white, and gaussian with known variance. The distributions of each of the 63 AC DCT coefficients are sometimes modeled by laplacian distributions [43,76]. In practice, the generalized gaussian is more accurate [9,89]. The DC coefficients are not cored because their conditional expectation depends too much on the specific sequence. The generalized gaussian distribution is given by:

$$P(x) = a \cdot \exp(-b \cdot |x|^c), \tag{5.20}$$

with:

$$a = \frac{b \cdot c}{2\Gamma(1/c)} \text{ and } b = \frac{1}{\sigma}\sqrt{\frac{\Gamma(3/c)}{\Gamma(1/c)}}, \tag{5.21}$$

where $\Gamma(\ )$ is the gamma function and σ is the standard deviation of the distribution. It can be seen from (5.20) and (5.21) that the generalized gaussian is completely determined by the shape parameter c and the noise variance σ^2. The well-known gaussian distribution is obtained by letting $c = 2$; the laplacian distribution is obtained by letting $c = 1$.

An efficient method for estimating the shape parameter c from a set of data based on second-order statistics is given in [89]. Let Y_k denote DCT coefficients with coefficient number $k = 1, 2, \ldots, 64$. The mean and the variance μ_k and σ_k^2 of a set of observed DCT coefficients with coefficient number k can be estimated directly from the observed data. Let ρ_k be:

$$\rho_k = \frac{\sigma_k^2}{E^2[|Y_k - \mu_k|]}. \tag{5.22}$$

The shape parameter c_k for the distribution of DCT coefficient k is found by solving:

$$\frac{\Gamma(1/c_k) \cdot \Gamma(3/c_k)}{\Gamma^2(2/c_k)} = \rho_k. \tag{5.23}$$

Equation (5.23) can be solved efficiently with a lookup table that is generated by letting c_k vary over the range of values that could possibly be expected for this parameter in small steps. Let c_k vary from 0.1 to 2.5 with a step size of, say, 0.01 for these steps. Then the generalized gaussian approximations to the distributions of the observed DCT coefficients are readily obtained from the c_k and the σ_k^2 with (5.21) and (5.20).

Figure 5.13 shows the c_k and the σ_k that are estimated from the DCT coefficients obtained from a set of 18 different images. The scanning order in a 2D block of DCTs is taken from left to right (increasing horizontal frequency) and from top to bottom (increasing vertical frequency); see Figure 5.14. Except for the first DCT coefficient, the DC component, it can be seen that c_k is a bit smaller than 0.5. The standard deviation of the coefficients decreases with increasing frequency, which is consistent with the well-known fact that natural images contain less energy in high frequencies than in low frequencies.

Figure 5.15 shows the coring function computed for DCT coefficient number 8 for noise with variance 100 corrupting the image. In this figure, small values are cored towards zero; larger values are altered less. This confirms the intuitive assumption that data with small amplitudes are noisy and unreliable, and they should therefore be discarded. Figure 5.16 plots the coring functions for all 64 DCT coefficients, again with noise with variance 100 corrupting the image. As already mentioned, the DC terms are not cored; hence the 45 degree line for this DCT coefficient. It can be seen that coefficients representing higher spatial frequencies are cored towards zero more strongly than coefficients representing lower spatial frequencies. This, again, matches well with the fact that natural images contain less energy in high frequencies than in low frequencies.

The coring functions depend on the noise variance. A number of lookup tables are computed for different noise variances in a practical situation. The MPEG2 encoder selects the lookup table that corresponds best with the actual noise variance in an image sequence.

5.4.3 Experiments and results

For the experiments, the standard *test model 5* (TM5) MPEG2 encoder [40] was adjusted so that the DCT coefficients are cored using lookup tables, as described in the previous sections. This section describes two sets of experiments. The same test sequences are used in Section 5.3.6.

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64

Figure 5.14 Numbering of DCT coefficients in an 8×8 block.

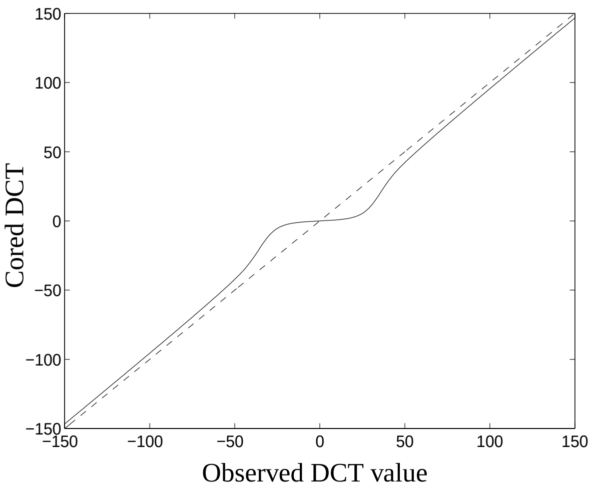


Figure 5.15 Coring function for DCT coefficient 8, computed for noise with variance 100 corrupting the image.

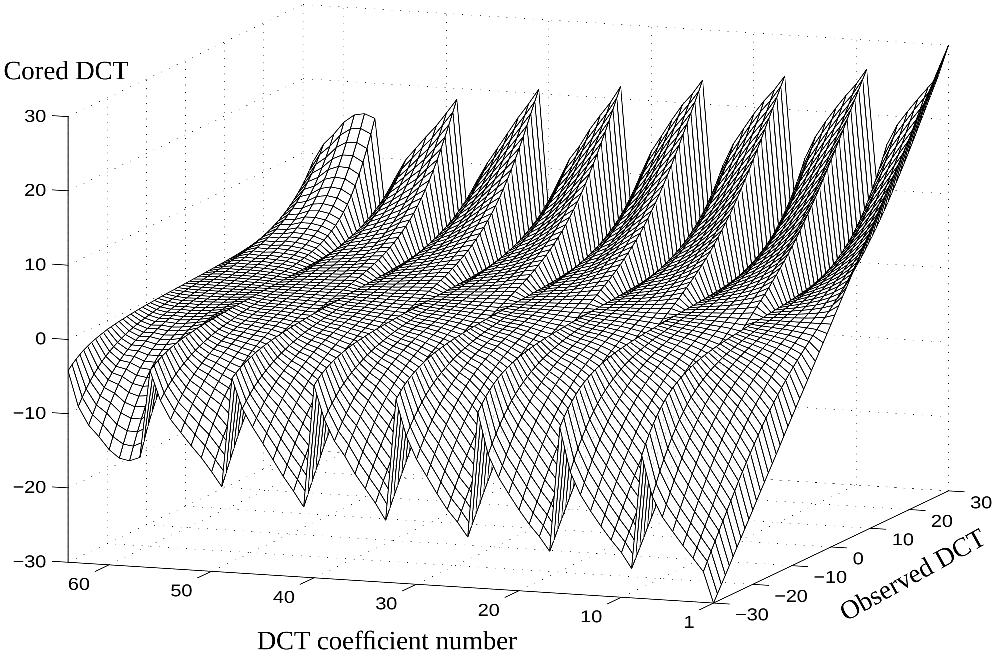
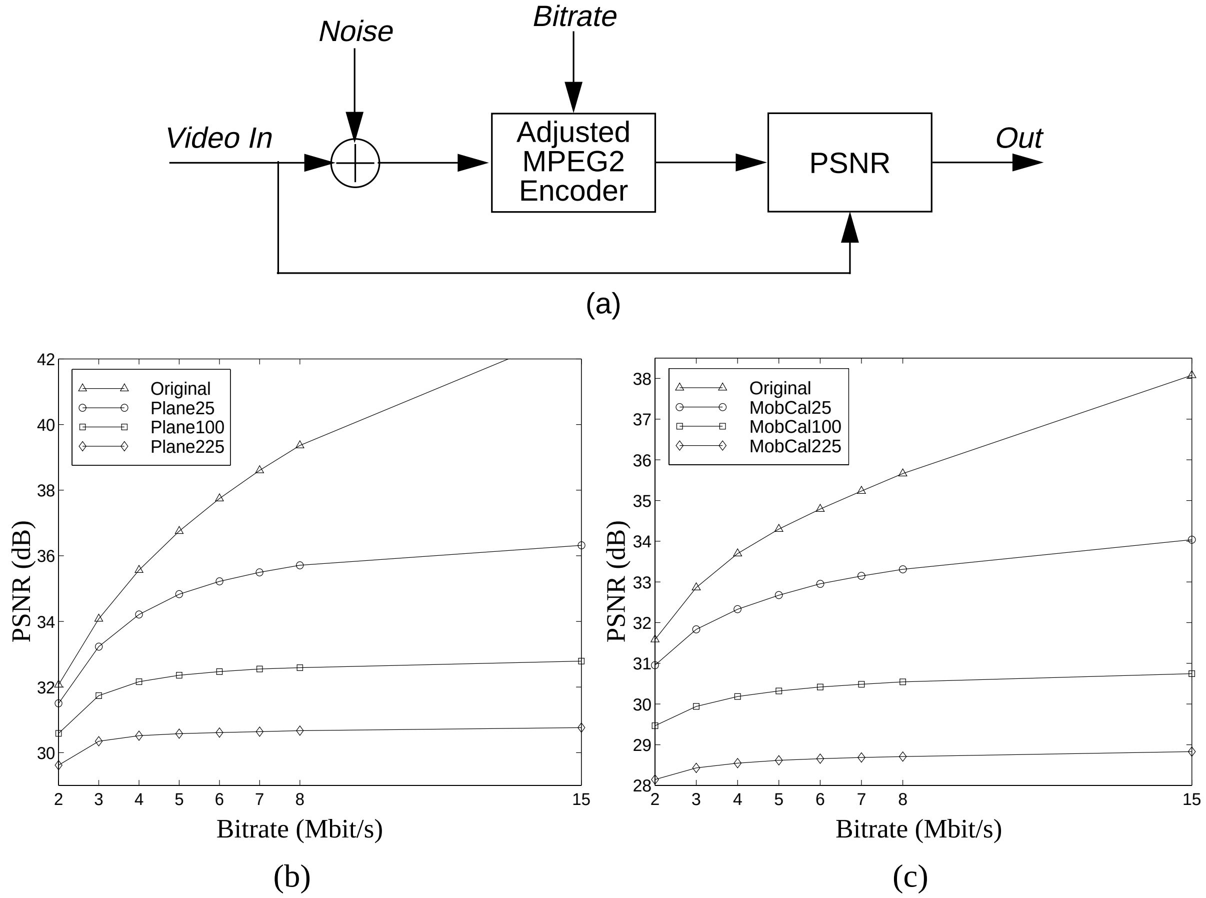


Figure 5.16 Plot of part of the coring functions for all 64 DCT coefficients, computed for noise with variance 100 corrupting the image.

The first set of experiments evaluates the performance of the adjusted TM5 encoder in terms of the PSNR when applied to test sequences with varying amount of noise. Figure 5.16a shows the scheme used for measuring the PSNR of the corrected sequences. Figure 5.16b,c plots the PSNRs for bitrates ranging from 2 Mbit/s to 15 Mbit/s. The results show that the PSNRs of the filtered and coded sequences are considerably higher at the higher bitrates than those of the noisy input sequences.

The PSNRs of the corrected sequences increase more rapidly with increasing bitrate at low bitrates than at high bitrates. Specifically, the curves for test sequence with noise variance 100



(a)

(b) (c)

Figure 5.16 (a) Scheme for measuring PSNRs of coded noisy test sequences. Results for (b) *Plane* and (c) *MobCal* sequences with the adjusted MPEG2 encoder and coring. The noise variance in the noisy sequences were 25, 100, and 225, which correspond to PSNRs of 33.0, 27.0, and 23.5 dB, respectively.

and 225 are quite flat over the range from 4 Mbit/s to 15 Mbit/s. This contrasts to the PSNRs for noise free sequences, which increase steadily with increasing bitrate. This implies that there is an “early” *saturation point* for the bitrate in noisy image sequences. Encoding with bitrates above the saturation point gives only marginal improvements in image quality.

By comparing the results in Figure 5.16, at for instance 8 Mbit/s, to those in Table 5.2, it can be seen that the 3D pyramid and wavelet filters outperform the adjusted MPEG2 encoder in terms of PSNR. However, the adjusted MPEG2 encoder is basically a 2D filter. It can be seen that its performance is similar to that of the 2D pyramid and 2D wavelet filters.

The second set of experiments investigates whether the adjusted TM5 encoder performs better than the standard encoder in combination with prefiltering, e.g., with the 3D pyramid noise-reduction system. It could be imagined that even though the 3D pyramid filter and the 3D wavelet filter on their own outperform the adjusted MPEG2 encoder, their superior quality may be lost due to quantization errors introduced by the standard encoder. Another question is how the performance of the adjusted MPEG2 encoder compares to the standard TM5 MPEG2 encoder with a dead zone when it is applied to a noisy sequence.

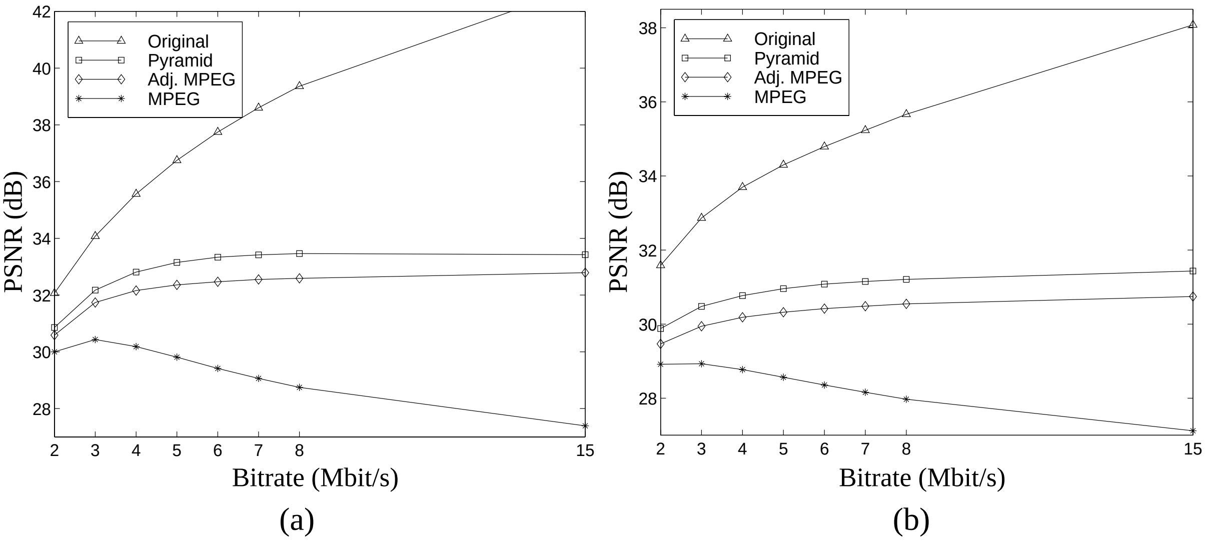


Figure 5.17 (a) PSNR vs. bitrate for the original *Plane* sequence, noisy *Plane* sequence (noise variance 100), and noise-reduced *Plane* sequence (filtered by the 3D pyramid filter) encoded by the standard TM5 MPEG2 encoder. Also shown is the PSNR of the noisy *Plane* sequence that was encoded and noise reduced simultaneously by the adjusted MPEG2 encoder with coring. (b) As before, but now for the *MobCal* sequence.

These questions are investigated, using the *Plane* and *MobCal* sequences to which a moderate amount of noise (variance 100) was added. Figure 5.17 plots the PSNRs as a function of the bitrate of the noisy test sequences after encoding by the standard TM5 with and without prefiltering by the 3D pyramid filter. The PSNRs that result from applying the adjusted TM5 coder to the noisy sequences are also shown. Finally, the PSNRs of the coded original, noise-free sequences are plotted as a reference of what can maximally be obtained.

Figure 5.17 indicates that prefiltering sequences with a moderate amount of noise prior to encoding with the standard TM5 encoder gives a PSNR that is maximally one decibel higher than when simultaneous filtering and encoding is done by the adjusted TM5 MPEG2 encoder. It can also be seen from Figure 5.17 that the standard TM5 encoder (without prefiltering) also functions as a noise reducer at low bitrates. At 3 Mbit/s, the PSNR of the coded noisy *Plane* sequence is about 3.5 dB higher than that of the noisy original. This number is 1.5 dB for the MobCal sequence. The PSNRs decrease for these sequences at higher bitrates. This behavior is not surprising. The encoder applies a coarse quantization at low bitrates and much noise energy is removed by the dead zone. The encoder is capable of encoding the signal and the noise more accurately at higher bitrates, so that the noise part of the signal is preserved better. In the limiting case, at very high bitrates, the noisy sequence is encoded without errors, and the PSNR equals 23.5 dB.

5.5 Discussion

This chapter shows that coring is a powerful technique for noise reduction. A 2D shift invariant wavelet filter and the 2D Simoncelli pyramid were introduced. These filters were extended in the temporal dimension so that temporal information, as well as spatial information, in image sequences could taken into account in the noise reduction process. The spatio-temporal decomposition allows temporal filtering of the DC bands of the 2D Simoncelli pyramid and the 2D DWT transforms without introducing severe blur or other artifacts. Two-dimensional scale-space noise reduction filters have no way of filtering the DC bands by means of coring.

The noise reduction capabilities of the Simoncelli pyramid outperforms those of the shift invariant DWT due to the minimal aliasing and its enhanced directional sensitivity. However, the difference in performance in terms of increase in PSNR can be considered marginal if one takes into account the increase in complexity for the pyramid filter compared to the wavelet filter.

Even though the 3D pyramid filter as presented in this chapter is a complex and expensive filter to implement, it is nevertheless a useful one. Visually speaking, the results obtained by the pyramid filter are better than those obtained from the shift invariant wavelet filter. It can be applied when good quality noise reduction is absolutely necessary, i.e., when processing time is less important than image quality. It can also be used as a benchmark for the results obtained by other filters.

This chapter also shows that the MPEG2 scheme can easily be adapted to perform simultaneous noise reduction and compression. The extra costs of the adapted scheme, compared to a standard MPEG2 encoder, consist of implementing lookup tables and an extra DCT operation for the B and P frames. This is a cheaper solution than the pyramid filter or the wavelet filter and it gives reasonable performance. In fact, the experiments indicate that, if a noisy image sequence is to be encoded, the difference between encoding the prefiltered sequence and encoding the noisy sequence with the adapted encoder is less than one decibel over a large range of bitrates. In this case, whether or not prefiltering is a cost-effective solution depends on the required quality of service.

Chapter 6

Evaluation of restored image sequences

6.1 Introduction

Chapter 1.1 explains the motivation for restoration of archived film and video. It is stated there that image restoration improves the perceived (subjective) quality of film and video sequences and that restoration also leads to more efficient compression. This chapter experimentally verifies the validity of these two assertions.

Section 6.2 describes the methodology that is used in two sets of experiments for validating the assumptions mentioned. The first set of experiments is aimed at verifying that image restoration indeed improves the perceived quality of impaired image sequences. These experiments are done with test panels. The second set of experiments is aimed at verifying that image restoration indeed improves the coding efficiency. This can be done with test panels, or, as is done in this chapter, by numerical evaluation. Section 6.3 describes and discusses the experimental results. Section 6.4 concludes this chapter, and, thereby, this thesis. It gives some pointers to future research.

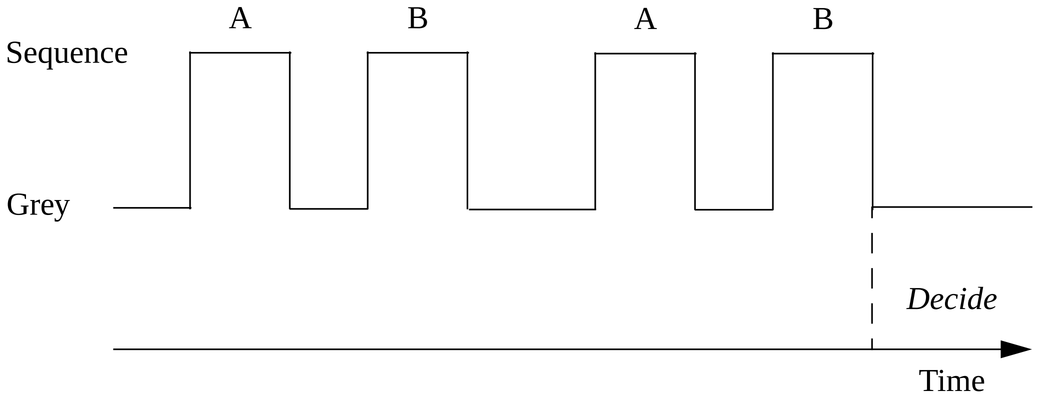


Figure 6.1 Overview of 2AFC testing.

6.2 Assessment of restored image sequences

6.2.1 Influence of image restoration on the perceived image quality

An important reason for image restoration is that it improves the image quality as perceived by humans. Whether the underlying assumption is indeed true can only be determined by having human observers compare restored sequences to the corresponding impaired sequences. So far, automatic validation (without human beings) is not possible: there are no mathematical models that can adequately model human perception of images in all their aspects.

The *International Telecommunication Union* (ITU) has standardized a number of methods for evaluating image sequences by test panels. For instance, the *double-stimulus continuous quality-scale* (DSCQS) method is well-known [41]. This method measures the relative difference in quality of an impaired sequence given the original, unimpaired image as a reference. The DSCQS method is useful for comparing the performance of various restoration systems. Evaluations using this method have been done in the AURORA project.

The scope of the evaluations in this chapter are not as broad as those in AURORA. At this point, the aim is not to compare the performance of different restoration systems. Here, the central question is whether the image restoration algorithms presented in this thesis improve the perceived image quality. A method simpler than the DSCQS method can be used for finding an answer to this question. The method used here is the *two alternatives forced choice* (2AFC) method [3]. The 2AFC method is often used in television broadcasting environments to determine at what point a transmission system introduces visible distortions in the transmitted images or image sequences. In the context of image restoration, this method is not used to determine whether there are visible differences between two sequences, but to determine which of the two sequences have the highest perceived quality.

In the 2AFC method, the members of a test panel are shown pairs of image sequences A and B twice, as illustrated by Figure 6.1. One of the sequences is the impaired sequence, the other is the restored sequence. Which is which is random. The duration of each sequence is approximately 10 seconds. Between showing sequences A and B, the screen is blanked to a mid-gray value for 2 seconds. After a pair of image sequences has been viewed for the first time, the screen is blanked to a mid-gray value for 5 seconds. Then, sequences A and B are shown again. If A was the impaired sequence in the first viewing, then it is also the impaired sequence in the second viewing. The same is true for B. After viewing the sequences the second time, the assessors must indicate which sequence has the better visual quality.

For all experiments in this chapter, differences between the impaired and the corrected sequences are clearly visible. The outcome of 2AFC testing is determined by one of two cases. In the first case, a majority of the votes is given to either A or B. This indicates a general consensus on whether the perceived quality of the corrected sequence is better than that of the impaired sequence. In the second case, about 50% of the votes is given to each of the sequences, and there is no general consensus on which sequence (impaired or restored) is better. The second case can occur, for example, in the case of noise reduction. It is well known that some people prefer a noisy image over a slightly blurred noise-free image. The noise gives an illusion of increased sharpness. Other people prefer a noise-reduced image, even if it is slightly blurred.

6.2.2 Influence of image restoration on the coding efficiency

This section describes experiments that can be carried out to verify that image restoration indeed does lead to more efficient image compression. Before it can be determined how much more efficient one image sequence is compressed with respect to another, a definition for the *increase in coding efficiency* is required.

Let ΔQ denote the increase in coding efficiency between a corrected image sequence and an impaired image sequence. ΔQ can be defined in two ways. The first definition relates ΔQ to the distortion introduced by a codec set to a fixed bitrate. The second definition relates ΔQ to the bandwidth required by a codec to compress a sequence given the allowable distortion.

ΔQ in terms of coding accuracy. Figure 6.2 proposes an experimental setup that can be used for measuring the increase in coding efficiency in terms of how accurately the corrected and the impaired image sequence are coded with respect to each other.

Let $\hat{y}_o(\boldsymbol{i})$ and $\hat{y}_c(\boldsymbol{i})$ be restored image sequences before and after coding, respectively. Similarly, let $z_o(\boldsymbol{i})$ and $z_c(\boldsymbol{i})$ be impaired image sequences before and after coding, respectively. In Figure 6.2, the restored image sequence is encoded at a constant bitrate. The PSNR computed between the codec input and output is given by $PSNR[\hat{y}_o, \hat{y}_c]$. The impaired image sequence is encoded at the same bitrate. In this case, the PSNR computed between codec input and decoded output is given by $PSNR[z_o, z_c]$. ΔQ is now defined by:

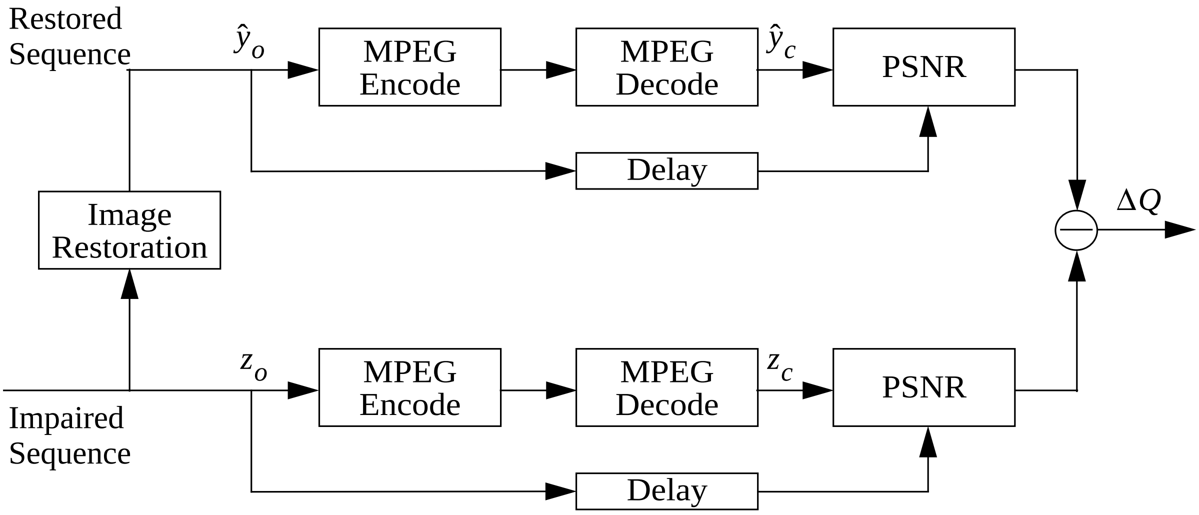


Figure 6.2 Method for measuring the difference in coding efficiency on the basis of PSNR.

$$
\begin{aligned}
\Delta Q &= PSNR[\hat{y}_o, \hat{y}_c] - PSNR[z_o, z_c] \\
&= 10 \cdot Log\left(\frac{224^2}{\frac{1}{N}\sum_{\boldsymbol{i}}(\hat{y}_o(\boldsymbol{i}) - \hat{y}_c(\boldsymbol{i}))^2}\right) - 10 \cdot Log\left(\frac{224^2}{\frac{1}{N}\sum_{\boldsymbol{i}}(z_o(\boldsymbol{i}) - z_c(\boldsymbol{i}))^2}\right) \\
&= 10 \cdot Log\left(\frac{\sum_{\boldsymbol{i}}(z_o(\boldsymbol{i}) - z_c(\boldsymbol{i}))^2}{\sum_{\boldsymbol{i}}(\hat{y}_o(\boldsymbol{i}) - \hat{y}_c(\boldsymbol{i}))^2}\right).
\end{aligned}
\tag{6.1}
$$

From (6.1) it can be seen that ΔQ is a function of the ratio between the energy of the coding errors in the impaired image sequence to the energy of the coding errors in the corrected image sequence. If $\Delta Q > 0$, the corrected sequence is coded with fewer errors than the impaired sequence. If $\Delta Q < 0$, the corrected sequence is more difficult to code than the impaired sequence and the compression errors are larger.

It is emphasized here that the coding errors in $\hat{y}_c(\boldsymbol{i})$ and $z_c(\boldsymbol{i})$ are computed between the input and output of the codec. The errors are not computed with respect to a ground truth, i.e., an unimpaired original. In practice, no unimpaired references exist for archived film and video material.

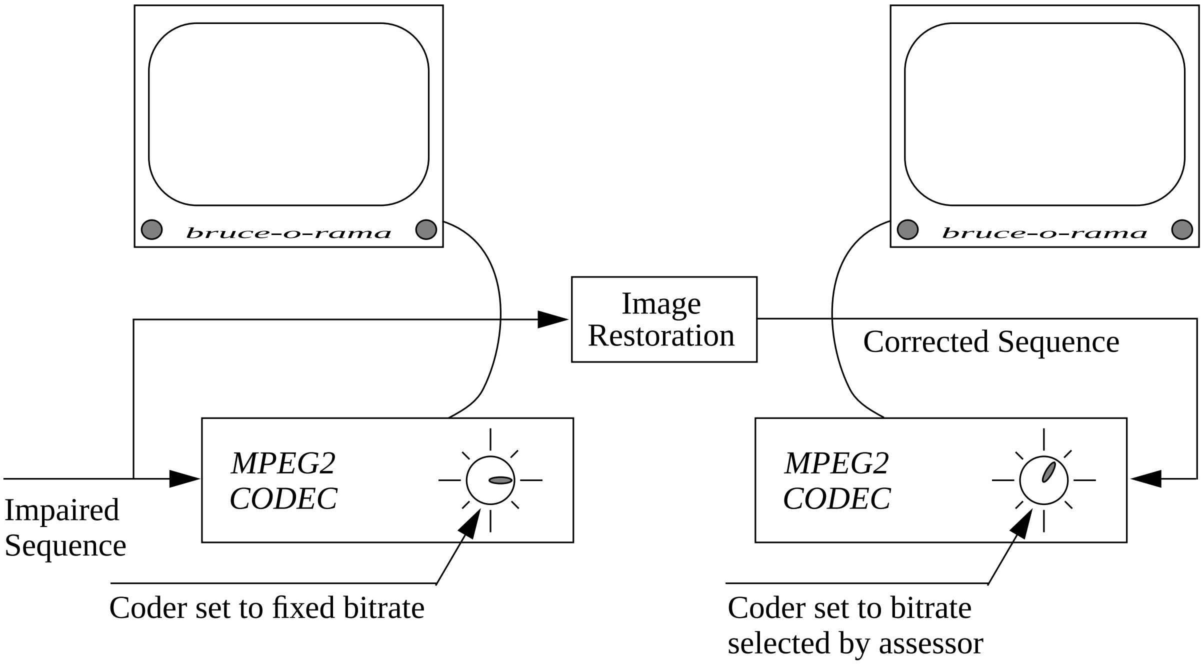


Figure 6.3 Setup for measuring the increase in coding efficiency using human assessors.

ΔQ in terms of bandwidth. The definition of ΔQ in terms of bandwidth is given by the difference in bitrate for the coded corrected image sequence and for the coded impaired image sequence:

$$\Delta Q = Bitrate[impaired] - Bitrate[corrected]. \tag{6.2}$$

Here, if $\Delta Q > 0$, the corrected sequence requires fewer bits for coding than the impaired sequence. If $\Delta Q < 0$, the corrected sequence is more difficult to code than the impaired sequence and it requires more bits. Obviously, ΔQ can only be given a meaningful interpretation if it is measured on condition that the bitrates selected for coding the impaired and corrected sequences are related in a meaningful way. The constraint set here for measuring (6.2) requires that the codec introduces the same amount of distortion to the impaired as to the corrected sequence.

This raises the question of how the distortion introduced by a codec should be measured. Ideally, the measured distortion is related to the perceived image quality. This requires involving human observers to determine (6.2) with, for instance, the setup proposed in Figure 6.3. In this setup, the impaired image sequence is coded by an MPEG2 codec set to a fixed bitrate. The impaired sequence is restored and coded by an MPEG2 codec of which the bitrate is controlled by an assessor. The codecs are synchronized to compensate for the delay introduced by the restoration system. Their outputs are displayed on two calibrated monitors. During the experiment, the task of the assessor is to set the bitrate of the codec he/she controls to a level such that the perceived quality of the coded corrected sequence is equal to that of the coded impaired sequence. The difference in bitrate of the two codecs gives the increase (or decrease) in coding efficiency given by the image restoration process.

Note that the type of artifacts in the coded impaired and corrected sequences can be different at the bitrates at which the assessor rates the perceived image quality the same. For instance, consider a noisy image sequence coded at a bitrate at which the codec does not introduce visible distortions. The corrected, noise free sequence can be coded at a lower bitrate. At a certain point this bitrate is so low that blocking artifacts start to appear. It is around this point that the assessor will begin to prefer the coded noisy sequence over the coded corrected sequence.

The method for measuring the improvement in coding efficiency with human assessors requires a fair amount of calibrated and synchronized equipment. An alternative method is to measure the distortion with mathematical measures based on the MSE. Obviously, the results will be different from those obtained by human assessors. In this case, a scheme similar to that in Figure 6.2 is used for measuring ΔQ. First, the corrected sequence is coded at a fixed bitrate. Next, the bitrate for coding the impaired sequence is searched so that $PSNR[\hat{y}_o, \hat{y}_c]$ equals $PSNR[z_o, z_c]$, i.e., so the same amount of compression errors have been introduced into the corrected and impaired sequence. Again, as in (6.2), ΔQ is given by the difference in bitrates.

As a final remark, it should be mentioned that ΔQ, measured either in dB or in Mbit/s, can only be meaningful if the restored image sequence consists of sensible data that represent the true image data in a reasonable manner. For example, it is assumed that the restored sequence is not a collection of black frames if the original data is clearly not a collection black frames, but, for instance, a recording of a zoo.

6.3 Experiments and results

This section experimentally verifies that the algorithms proposed in this thesis indeed improve the perceived image quality by presenting the impaired and restored image sequences to a test panel. The influence of image restoration on the perceived image quality is assessed in two circumstances. In the first circumstance, pairs of impaired and corrected sequences are used. In the second circumstance, pairs of MPEG2 encoded impaired and corrected sequences are used. The latter circumstance verifies the assumption that image restoration improves the perceived image quality also holds in a digital broadcasting environment.

This section also verifies that the algorithms developed in this thesis improve the coding efficiency. The increase in coding efficiency, ΔQ, is determined by numerical evaluation, both in terms of PSNR and in terms Mbit/s.

6.3.1 Test sequences

To get an impression of the effects of removing different combinations of artifacts on the perceived image quality and on the increase in coding efficiency, test sequences were selected

Sequence	Amount of Flicker	Number of Blotches	Visibility of Noise
Plane (100 Frames)	High	High	High
Chaplin (112 Frames)	Medium	High	Medium
Charlie (48 Frames)	High	High	Low
Mine (404 Frames)	Medium	Very Low	High
VJ Day (49 Frames)	None	Low/Medium	Low
Soldier (227 Frames)	Medium	Very Low	Low

Table 6.1 List of impaired sequences used for subjective and objective evaluations with an indication of the severity of the various degradations. Note that the *Plane* sequence contains artificial degradations.

Sequence	Flicker Correction	Blotch Correction	Noise Reduction
Plane	X	X	X
Chaplin	X	X	X
Charlie	X	X	
Mine	X		X
VJ Day		X	
Soldier	X		

Table 6.2 Corrections applied to the test sequences.

with various combinations of impairments. The test sequences consist of one artificially degraded sequence and five naturally degraded sequences. Table 6.1 lists the sequences and gives an indication of the severity of the degradations that impair them. The test sequences are also used in Chapters 3 to 5 and have already been described. An exception is the *Chaplin* sequence, which has not been used before for any experiment. Three frames from this sequence are shown in Chapter 1, Figure 1.1.

Table 6.2 lists the artifacts that were corrected in each of the test sequences by the restoration system depicted in Figure 1.2 with the restoration methods developed in this thesis. The various control parameters of the restoration algorithms were set to values that give good visual results.

6.3.2 Experiments on image restoration and perceived quality

The subjective experiments were done in a dimly lit room. The viewing distance was six times the height of the display used. The test panel consisted of 25 people, all of whom had good vision with a visus of 0.8 or better. Before the actual experiments, the assessors were trained for their task by being shown some examples of sequences with and without flicker, blotches, and noise. Each assessor assessed all the test sequences once. They were asked the following question: “Which sequence do you find more pleasing to view, A or B?”.

As already mentioned, each test sequence should be approximately 10 seconds in duration. Because most of the test sequences are shorter than 10 seconds, they were repeated (looped) a number times so that the duration of the looped sequence was approximately 10 seconds. Only the first 10 seconds of the 16-second *Mine* sequence were shown.

Table 6.3 gives the results for the first set of experiments in which the assessors indicated which sequence they prefer: the impaired image sequence or the restored image sequence. This table shows that, for all test sequences, the majority of the votes is given to restored image sequences. This proves that the image restoration algorithms presented in this thesis increase the perceived image quality of impaired image sequences.

The restored *Mine* sequence received relatively fewer votes than the other restored sequences. When questioned about this, some of the test panel members indicated they considered the corrected *Mine* sequence to be overly smooth, and, therefore, they preferred the flickering, noisy original. The smoothing was caused by the noise reduction algorithm that was set to achieve a great amount of noise reduction. It is a well-known fact that there is a trade-off between noise reduction and introducing blur. Had the noise reducer been set for less noise reduction, less smoothing would have been introduced, and the assessors in question might well have preferred the corrected sequence.

Table 6.4 gives the results for the second set of experiments in which the assessors indicate which sequence they prefer: the MPEG2 encoded impaired image sequence or the MPEG2 encoded restored image sequence. The standard TM5 MPEG2 encoder was used for all experiments [40]. The coder was set to the main profile and the GOP size was 12. This table shows that for all test sequences, the majority of the votes is given to MPEG2 encoded restored image sequences. This proves that the increase in perceived quality, obtained from the image restoration algorithms presented in this thesis, are not lost due to coding artifacts introduced by an MPEG2 encoder at 4 Mbit/s. Therefore, image restoration is beneficial in digital broadcasting environments in which films are broadcast in compressed format.

6.3.3 Experiments on image restoration and coding efficiency

This section presents the results of two sets of numerical evaluations. The first set applies the scheme shown in Figure 6.2 to measure the increase coding efficiency in dB. ΔQ was evaluated for bitrates ranging from 2 Mbit/s to 8 Mbit/s. For all experiments the standard TM5 MPEG2 encoder was used. The coder was set to the main profile and the GOP size was 12.

Sequence	Votes for Corrected Sequence (in percentages)	Votes for Impaired Sequence (in percentages)
Plane	84	16
Chaplin	92	8
Charlie	84	16
Mine	72	28
VJ Day	88	12
Soldier	92	8

Table 6.3 Results of subjective evaluations for the first set of experiments in which impaired and restored sequences are compared.

Sequence	Votes for Corrected Sequence (in percentages)	Votes for Impaired Sequence (in percentages)
Plane	88	12
Chaplin	100	0
Charlie	88	12
Mine	68	32
VJ Day	80	20
Soldier	96	4

Table 6.4 Results of subjective evaluations for the second set of experiments in which impaired and restored sequences are compared after MPEG compression at 4 Mbit/s

Figure 6.4 plots the results for this first set of experiments. The curves indicate that image restoration leads to more efficient compression over the range of investigated bitrates; at identical bitrates, the restored image sequences can be compressed with fewer errors than the impaired sequences. This proves that image restoration gives more efficient compression for the artifacts considered.

The gains are smallest for the *VJ Day* sequence. Only the blotches were restored in this sequence. Because the blotches cover only a small percentage of the total image area in this sequence, removing them has little influence on the overall coding efficiency. The gains for the *Soldier* sequence, which was corrected for intensity flicker, are somewhat larger. The intensity flicker is a global effect and has a larger influence on the coding efficiency. The *Charlie* sequence contained much flicker and many blotches. Restoring this sequence gives large gains.

The restored *Plane*, *Chaplin*, and *Mine* sequences give the largest increases in coding efficiency. Unlike the other test sequences, these sequences were noise reduced. Noise is difficult to code and removing it simplifies the coder's task (unless, of course, the adjusted coder described in Chapter 5 is used). ΔQ is largest for the *Mine* sequence. As mentioned in the previous section, the corrected *Mine* sequence is quite smooth. Hence it can be coded with many fewer errors than the impaired original.

The second set of experiments in this section measures ΔQ in terms of bandwidth, i.e., in Mbit/s. At the time the experiments were carried out, the equipment for measuring ΔQ with human assessors, as described in Section 6.2.2, was not available. The numerical method, also described in Section 6.2.2, was used.

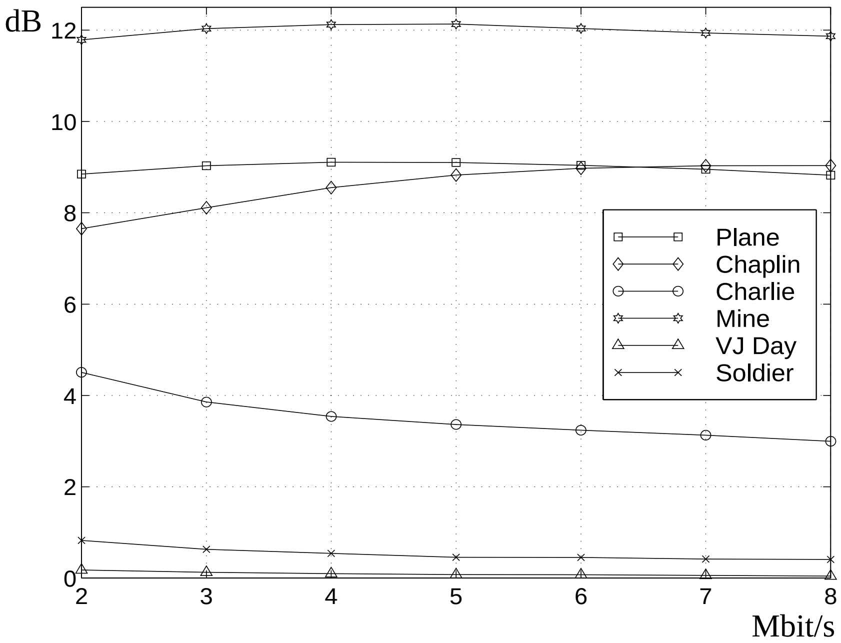


Figure 6.4 ΔQ measured in *dB* versus bitrate.

Sequence	PSNR of $\hat{y}(\boldsymbol{i})$ at 4 Mbit/s (in dB)	Bitrate for $z(\boldsymbol{i})$ with same PSNR (in Mbit/s)	ΔQ (in Mbit/s)	Savings in Bandwidth by Restoration (in percentages)
Plane	36.4	15.5	11.5	74,2
Chaplin	39.1	19.9	15.9	79.9
Charlie	40.1	9.0	5.0	55.6
Mine	44.3	38	34.0	89.5
VJ Day	34.0	4.2	0.2	4.8
Soldier	36.8	4.8	0.8	16.7

Table 6.5 Results of numerical evaluations of ΔQ measured in Mbit/s.

The experiment was set up as follows. First, the PSNR ratios were computed over the encoded/decoded *restored* image sequences coded at 4 Mbit/s (broadcast quality). Next, the impaired sequences were coded at bitrates so that the PSNRs over the coded/decoded *impaired* sequences were identical to those of the corrected sequences. The differences in bitrate gives the increase in coding efficiency. The standard TM5 MPEG2 encoder was used for all experiments. The coder was set to the *main* profile or, for bitrates greater than 15 Mbit/s, to the *high* profile, and the GOP size was 12.

Table 6.5 lists the results from the second set of experiments. Again, it is concluded that image restoration leads to more efficient compression. Considerable savings in bandwidth can be achieved by restoring impaired image sequences. Again, the largest gains were obtained for the test sequences to which noise reduction was applied. The last column in this table was computed by:

$$percentage = \frac{Bitrate[z(\boldsymbol{i})] - 4}{Bitrate[z(\boldsymbol{i})]} \times 100\ \%. \tag{6.3}$$

6.3.4 Discussion of experimental results

The experimental results verify that the image restoration algorithms developed in this thesis improve the perceived image quality of impaired image sequences. The experiments also verify that image restoration improves the coding efficiency. Therefore, the benefits of restoration of archived film and video is confirmed, and the assumptions underlying the work carried out in this thesis are validated.

A question is how well the numerical experiments for determining the increase in coding efficiency correspond to human perception. ΔQ, as defined in this chapter, reflects an increase in image quality terms of PSNR or in terms of how many bits of irrelevancy have been removed. It is well known that numerical measures do not necessarily correlate well with subjective perception. For instance, ΔQ is a global measure, whereas human observers are very sensitive to local variations in image quality. An example that illustrates this is given by the experimental results for the *VJ Day* sequence. This sequence was corrected for local artifacts, namely blotches. The results from the test panel evaluation shows that a majority of 88% prefers the corrected sequence over the impaired sequence. The large number of votes implies a clearly visible improvement in the perceived image quality. In contrast, the ΔQ computed for this sequence in terms of PSNR and in terms of bandwidth are small; 0.1 dB and 0.2 Mbit/s, respectively. Therefore, they suggest a marginal improvement only.

6.4 Discussion

This thesis presented new methods for image restoration that have proven to be very successful. Is there still room for improvement? The answer is: yes. Automated image restoration is in fact still in its infancy, for three reasons. The first reason is related to the fact that, so far, there is no numerical measure for image quality that adequately models human perception. Human intervention for setting key parameters in the restoration system to values that give optimal results, visually speaking, is still necessary with the current technology. For example, the operator controls the amount of noise reduction to avoid overly smooth results. He or she detects and corrects instances in which a blotch detection and correction system fails. The image restoration processes are not completely automated in the true sense of the word.

The second reason why automated image restoration can be considered to be in its infancy is that the restoration techniques presented in this thesis are basically pixel based methods. The restoration algorithms operate on pixels with their local temporal and spatial neighborhoods; it is assumed that the local intensities follow certain statistical models. Often the parameters for this statistical model are inferred from large data sets and they do not necessarily reflect the true local statistics anywhere in an image or image sequence. This, of course, is suboptimal.

The third reason is that image restoration techniques rely heavily on accurate motion estimation and compensation. To date, motion estimators have difficulty coping with non-rigid motion and illumination variations. This is not beneficial to the performance of image sequence restoration algorithms.

If one tries to predict how the problems mentioned can be overcome, it is useful to draw parallels with the field of image coding. Image restoration and image coding are intimately related in the sense that they exploit temporal and spatial redundancy to isolate the essence of images and in that they discard irrelevancy.

In the image-coding society, much attention is given to developing objective, numerical measures for the perceived quality of images and image sequences that correlate well to human perception [99,102]. Currently, most of the research is focussed on measuring the influence of typical coding artifacts (e.g., blocking artifacts and blur) on the perceived image quality. These methods must be extended to artifacts common in old film and video sequences. A great challenge in this area is that, unlike for the case of video coding, no unimpaired references are available to serve as ground truths.

The MPEG4 standard initiates a trend towards object-based image coding. For MPEG4 to be used to the fullest of its potential, algorithms capable of meaningful image segmentation will have to be developed. Image restoration algorithms can exploit these segmentation results. The image restoration problem then shifts from pixel-based to region-based processing. Spatial and temporal correlations within corresponding regions can be exploited to get better estimates of local image statistics.

In conclusion, image segmentation results can also be used by restoration algorithms for higher level reasoning. For example, objects with large deformations from frame to frame pose severe problems to blotch detectors. Many false alarms result, for instance, from a bird that is flapping its wings rapidly in the process of flying. By relating the segmentation results for a number of frames, a smooth motion trajectory may be found for segments that define that bird. This implies temporal consistency; those segments do not represent blotches. Higher-level reasoning about the image contents can also be done by motion estimators to make them more robust to object deformations and to illuminance variations. This in turn is also beneficial to image sequence restoration.

Appendix A

Hierarchical motion estimation

Full-search block matching is a well-known method for estimating motion from a source frame to a reference frame. In this method, the source frame is subdivided into image blocks of 8×8 or 16×16 pixels. An exhaustive search is performed for each image block to find the optimal match within the reference frame. The *summed squared difference* (SSD) and the *summed absolute difference* (SAD) are often used as matching criteria. The displacement that gives the optimal match yields the motion estimate [31, 93].

Full search block matching is very intensive from a computational point of view. Furthermore, the motion vectors obtained from this technique do not necessarily represent (a projection onto two dimensions) of the true motion. They merely represent displacements that give optimal matches.

A method that suffers less from these drawbacks is hierarchical block matching [11,31,93]. Figure A.1 shows the principle of this method. First, initial, coarse motion vectors are estimated by applying (full-search) block matching to subsampled images. Next, the initial motion estimates are propagated to the next level with higher resolution and refined. Instead of full-search block matching, the refinements consist of doing a limited search in the region centered around the initial, coarse motion estimate. Again, the refined motion vectors are then propagated to the next level with higher resolution. The refinement process is repeated until the motion vectors have been computed for the source image at full resolution.

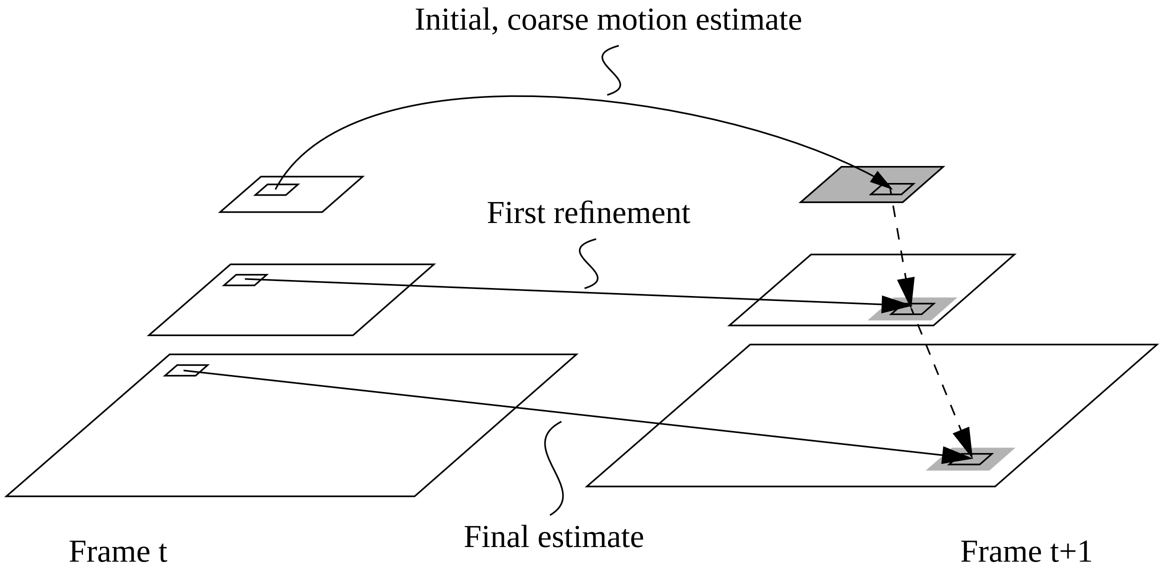


Figure A.1 General principle of hierarchical block matching. The gray areas indicate the search region for the block matching process.

As a result of the subsampling and the limited search strategies, hierarchical block matching requires fewer computations than full search block matching. Therefore it is faster. Furthermore, the final hierarchical motion estimates are closer to the true motion and the motion vectors are more consistent locally than the full search block matching motion estimates.

The reason for this is that the initial motion estimates are done on coarse images. An 8×8 image region in an image subsampled horizontally and vertically by a factor 4 corresponds to an image region of 32×32 in the high-resolution image. Therefore, the initial motion estimates computed by the hierarchical block matcher take more context into account than a full-search block matcher that uses 8×8 image blocks. Because motion estimates are propagated from coarse resolution levels to finer resolution levels, the refined estimates for adjacent blocks in the higher resolution images are made on the basis of the same initial vectors. Therefore, the final motion estimates are consistent locally.

As is explained in Chapter 2, hierarchical motion estimators are relatively robust to common artifacts in video and film sequences.

Appendix B

Derivation of conditionals

B.1 Introduction

Section 4.5.2 stated that draws have to be taken from the conditionals:

$$
\begin{aligned}
&\boldsymbol{a} \sim P[\boldsymbol{a}|\sigma_e^2, o(\boldsymbol{i}), z_+, d],\\
&\sigma_e^2 \sim P[\sigma_e^2\big|\boldsymbol{a}, o(\boldsymbol{i}), z_+, d],\\
&o(i) \sim P[o(i)|\boldsymbol{a}, \sigma_e^2, z_+(\boldsymbol{i}), d(\boldsymbol{i}), O].
\end{aligned}
\tag{B.1}
$$

This section shows that in the case of drawing samples from a conditional, it is not necessary that the conditional be known exactly. It suffices that the distribution of the samples follows a function that is proportional to the conditional. The following sections derive such functions for the conditionals in (B.1).

Bayes' rule states:

$$
P[A|B] = \frac{P[B|A] \cdot P[A]}{P[B]}. \tag{B.2}
$$

The goal is to draw samples for A, given B, from $P[A|B]$. Because B is given, $P[B]$ can be regarded as a normalizing constant. It is therefore only necessary that the draw for A be proportional to:

$$P[A|B] \propto P[B|A] \cdot P[A] . \tag{B.3}$$

This means that, when deriving expressions for conditionals from which samples are to be drawn, it is not necessary to compute the normalizing constants. Let B indicate a collection of random variables, $b_1, b_2, \ldots, b_n$, and suppose that b_1 is independent of A. Then:

$$\begin{aligned} P[A|B] &\propto P[b_1, b_2, \ldots, b_n|A] \cdot P[A] \\ &= P[b_1|b_2, \ldots, b_n] \cdot P[b_2, \ldots, b_n|A] \cdot P[A] \\ &\propto P[b_2, \ldots, b_n|A] \cdot P[A]. \end{aligned} \tag{B.4}$$

It can be seen from (B.4) that it is only necessary to consider terms involving A for drawing random samples for A, given B.

Before deriving the conditionals in (B.1), first a quick word about notation. In this section, bold faced characters describe matrices (capital letters) or vectors (small letters). For instance, $\boldsymbol{z}$ represents a vector into which an observed frame $z(\boldsymbol{i})$ has been scanned in a lexicographic fashion. Analogous to Chapter 4, $\boldsymbol{z}_+$ indicates a vector containing the motion-compensated previous, current, and next frame.

B.2 Conditional for AR coefficients

Each image region with missing data is modeled by a 2D AR process that uses a single set of coefficients $\boldsymbol{a}$. The conditional for $\boldsymbol{a}$ is given by:

$$P[\boldsymbol{a}|\sigma_e^2, \boldsymbol{o}, \boldsymbol{z}_+, \boldsymbol{d}] = \frac{P[\boldsymbol{a}, \sigma_e^2, \boldsymbol{o}|\boldsymbol{z}_+, \boldsymbol{d}]}{\int_{\boldsymbol{a}} P[\boldsymbol{a}, \sigma_e^2, \boldsymbol{o}|\boldsymbol{z}_+, \boldsymbol{d}] d\boldsymbol{a}} . \tag{B.5}$$

At first glance this might seem to be a very complex distribution. Fortunately, as is shown in [47], it turns out that (B.5) is proportional to a multivariate gaussian distribution. The derivation of [47] is repeated here.

First, it is noted that the denominator in (B.5) is independent of $\boldsymbol{a}$ and hence it can be considered as a normalizing constant that can safely be ignored. Therefore:

$$
\begin{aligned}
P[\boldsymbol{a}|\sigma_e^2, \boldsymbol{o}, \boldsymbol{z}_+, \boldsymbol{d}] &\propto P[\boldsymbol{a}, \sigma_e^2, \boldsymbol{o}|\boldsymbol{z}_+, \boldsymbol{d}] \\
&\propto P[\boldsymbol{z}_+|\boldsymbol{a}, o, \sigma_e^2, \boldsymbol{d}] \cdot P[\boldsymbol{a}] \\
&\propto \frac{1}{(2\pi\sigma_e^2)^{N/2}} \exp\left(\frac{-\boldsymbol{e}^T\boldsymbol{e}}{2\sigma_e^2}\right) \cdot P[\boldsymbol{a}],
\end{aligned}
\tag{B.6}
$$

where the identity in (4.33) has been used. Note that (4.33) applies to single pixels whereas (B.6) applies to image blocks with N pixels. Hence the factor N in the last line of (B.6). In this equation, $\boldsymbol{e}$ is a vector with prediction errors and T indicates the transpose operator. This prediction error vector is given by reformulating (4.30) in vector-matrix notation:

$$
\begin{aligned}
\hat{\boldsymbol{y}} &= \boldsymbol{A}\hat{\boldsymbol{y}} + \boldsymbol{e} \\
&= \hat{\boldsymbol{Y}}\boldsymbol{a} + \boldsymbol{e}.
\end{aligned}
\tag{B.7}
$$

The top line in (B.7) gives the usual vector-matrix representation of an AR image model in which a sparse matrix $\boldsymbol{A}$ that contains the prediction coefficients is multiplied with an image vector. Here, for convenience, the definition in the bottom line in (B.7) is used where the AR coefficients are placed in $\boldsymbol{a}$, and $\hat{y}(\boldsymbol{i})$ is scanned into matrix $\hat{\boldsymbol{Y}}$ such that $\hat{\boldsymbol{Y}}\boldsymbol{a} = \boldsymbol{A}\hat{\boldsymbol{y}}$.

The term $\boldsymbol{e}^T\boldsymbol{e}$ in (B.6) is examined more closely now:

$$
\begin{aligned}
\boldsymbol{e}^T\boldsymbol{e} &= (\hat{\boldsymbol{y}} - \hat{\boldsymbol{Y}}\boldsymbol{a})^T \cdot (\hat{\boldsymbol{y}} - \hat{\boldsymbol{Y}}\boldsymbol{a}) \\
&= \hat{\boldsymbol{y}}^T\hat{\boldsymbol{y}} - 2\hat{\boldsymbol{y}}^T\hat{\boldsymbol{Y}}\boldsymbol{a} + \boldsymbol{a}^T\hat{\boldsymbol{Y}}^T\hat{\boldsymbol{Y}}\boldsymbol{a} \\
&= (\boldsymbol{a} - (\hat{\boldsymbol{Y}}^T\hat{\boldsymbol{Y}})^{-1}\hat{\boldsymbol{Y}}^T\hat{\boldsymbol{y}}) \cdot (\hat{\boldsymbol{Y}}^T\hat{\boldsymbol{Y}}) \cdot (\boldsymbol{a} - (\hat{\boldsymbol{Y}}^T\hat{\boldsymbol{Y}})^{-1}\hat{\boldsymbol{Y}}^T\hat{\boldsymbol{y}}) + \hat{\boldsymbol{y}}^T\hat{\boldsymbol{y}} - \hat{\boldsymbol{y}}^T\hat{\boldsymbol{Y}}(\hat{\boldsymbol{Y}}^T\hat{\boldsymbol{Y}})^{-1}\hat{\boldsymbol{Y}}^T\hat{\boldsymbol{y}}.
\end{aligned}
\tag{B.8}
$$

Substituting those terms in (B.8) that involve $\boldsymbol{a}$ into (B.6), and also keeping in mind that $P(\boldsymbol{a})$ has a uniform distribution assigned to it, i.e., that it is a constant, gives:

$$
\begin{gathered}
P[\boldsymbol{a}|\sigma_e^2, \boldsymbol{o}, \boldsymbol{z}_+, \boldsymbol{d}] \propto \\
\frac{1}{(2\pi\sigma_e^2)^{N/2}} \exp\left(-\frac{(\boldsymbol{a} - (\hat{\boldsymbol{Y}}^T\hat{\boldsymbol{Y}})^{-1}\hat{\boldsymbol{Y}}^T\hat{\boldsymbol{y}}) \cdot (\hat{\boldsymbol{Y}}^T\hat{\boldsymbol{Y}}) \cdot (\boldsymbol{a} - (\hat{\boldsymbol{Y}}^T\hat{\boldsymbol{Y}})^{-1}\hat{\boldsymbol{Y}}^T\hat{\boldsymbol{y}})}{2\sigma_e^2}\right).
\end{gathered}
\tag{B.9}
$$

This can be recognized as proportional to a multivariate gaussian and can be denoted compactly as:

$$P[\boldsymbol{a}|\hat{\boldsymbol{y}}, \sigma_e^2, \boldsymbol{o}, \boldsymbol{z}_+, \boldsymbol{d}] \propto N(\hat{\boldsymbol{a}}, \sigma_e^2 \cdot (\hat{\boldsymbol{Y}}^T\hat{\boldsymbol{Y}})^{-1}), \tag{B.10}$$

where $\hat{\boldsymbol{a}} = (\hat{\boldsymbol{Y}}^T\hat{\boldsymbol{Y}})^{-1}\hat{\boldsymbol{Y}}^T\hat{\boldsymbol{y}}$ is the least squares estimate for the AR coefficients. $\hat{\boldsymbol{Y}}^T\hat{\boldsymbol{Y}}$ and $\hat{\boldsymbol{Y}}^T\hat{\boldsymbol{y}}$ can be recognized as estimates for the autocorrelation matrix $\boldsymbol{R}_{\hat{y}\hat{y}}$ and the autocorrelation vector $\boldsymbol{r}_{\hat{y}\hat{y}}$. These are necessary for solving the normal equations [53,94]. The pdf for $\boldsymbol{a}$ is thus shown to be proportional to a well-known distribution.

B.3 Conditional for the prediction error variance

A single error variance parameter σ_e^2 is associated with each image region with missing data. The conditional for σ_e^2 is given by:

$$P[\sigma_e^2|\boldsymbol{a}, \boldsymbol{o}, \boldsymbol{z}_+, \boldsymbol{d}] = \frac{P[\boldsymbol{a}, \sigma_e^2, \boldsymbol{o}|\boldsymbol{z}_+, \boldsymbol{d}]}{\int\limits_{\sigma_e^2} P[\boldsymbol{a}, \sigma_e^2, \boldsymbol{o}|\boldsymbol{z}_+, \boldsymbol{d}]d\sigma_e^2}. \tag{B.11}$$

Again, the denominator can be viewed as a normalizing constant that can safely be ignored:

$$\begin{aligned} P[\sigma_e^2|\boldsymbol{a}, \boldsymbol{o}, \boldsymbol{z}_+, \boldsymbol{d}] &\propto P[\boldsymbol{a}, \sigma_e^2, \boldsymbol{o}|\boldsymbol{z}_+, \boldsymbol{d}] \\ &\propto P[\hat{\boldsymbol{y}}|\boldsymbol{a}, \sigma_e^2] \cdot P[\sigma_e^2] \\ &= \frac{1}{(2\pi\sigma_e^2)^{N/2}} \exp\left(\frac{-\boldsymbol{e}^T\boldsymbol{e}}{2\sigma_e^2}\right) \cdot P[\sigma_e^2]. \end{aligned} \tag{B.12}$$

In [47] an equation is derived that is very similar to that in (B.12) and it is noted there that the result is proportional to an inverted gamma distribution $IG(x|\psi, \omega)$ with parameters ψ and ω:

$$IG(x|\psi, \omega) = \frac{\omega^\psi}{\Gamma(\psi) \cdot x^{\psi+1}} \exp\left(-\frac{\omega}{x}\right). \tag{B.13}$$

If $x = \sigma_e^2$, $\psi = N/2$, and $\omega = \boldsymbol{e}^T\boldsymbol{e}/2$, then (B.12) is proportional to (B.13), which means that:

$$P[\sigma_e^2|\boldsymbol{a}, \boldsymbol{o}, \boldsymbol{z}_+, \boldsymbol{d}] \propto IG\left(\sigma_e^2\middle|\frac{N}{2}, \frac{\boldsymbol{e}^T\boldsymbol{e}}{2}\right). \tag{B.14}$$

B.4 Conditional for the direction of interpolation

Unlike the AR model parameters, the direction of interpolation is computed on a pixel-by-pixel basis instead of on a block-by-block basis. The conditional for $o(\boldsymbol{i})$ is derived here. At each particular site $\boldsymbol{i}$ the conditional is given by:

$$P[o(\boldsymbol{i})|\boldsymbol{a},\sigma_e^2,\boldsymbol{z}_+,\boldsymbol{d},O] = \frac{P[\boldsymbol{a},\sigma_e^2,o(\boldsymbol{i})|\boldsymbol{z}_+,\boldsymbol{d},O]}{\int_{\boldsymbol{o}} P[\boldsymbol{a},\sigma_e^2,\boldsymbol{o}|\boldsymbol{z}_+,\boldsymbol{d},O]\,d\boldsymbol{o}}. \tag{B.15}$$

Here O indicates the direction of interpolation for the pixels in the local region surrounding $o(\boldsymbol{i})$. Collecting those terms that are proportional to the variables of interest gives:

$$\begin{aligned}
&P[o(\boldsymbol{i})|\boldsymbol{a},\sigma_e^2,\boldsymbol{z}_+,\boldsymbol{d},O]\\
&\propto P[z_+(\boldsymbol{i})\big|\boldsymbol{a},o(\boldsymbol{i}),\sigma_e^2,\boldsymbol{d}]\cdot P[o(\boldsymbol{i})|O]\\
&\propto \exp\left(-\frac{-e^2(\boldsymbol{i})}{2\sigma_e^2}\right)\cdot\exp\left(-\sum_k \beta\left|o(\boldsymbol{i})-o(\boldsymbol{i}+\boldsymbol{q}_k)\right|\right)\\
&\propto \exp\Bigg(-\sum_{\boldsymbol{i}\in S}[(1-d(\boldsymbol{i}))\cdot(z(\boldsymbol{i})-AR(\hat{\boldsymbol{y}},\boldsymbol{i},\boldsymbol{a}))^2 +\\
&\qquad d(\boldsymbol{i})\cdot(o(\boldsymbol{i})\cdot z_{mc}(\boldsymbol{i},t+1)+(1-o(\boldsymbol{r}))\cdot z_{mc}(\boldsymbol{i},t-1)-AR(\hat{\boldsymbol{y}},\boldsymbol{i},\boldsymbol{a}))^2]+\\
&\qquad \sum_k \beta\left|o(\boldsymbol{i})-o(\boldsymbol{i}+\boldsymbol{q}_k)\right|]\Bigg).
\end{aligned} \tag{B.16}$$

As in Chapter 4, $AR(\hat{y},\boldsymbol{a},\boldsymbol{i})$ denotes the prediction of the corrected image $\hat{y}$ at location $\boldsymbol{i}$. $AR(\hat{y},\boldsymbol{a},\boldsymbol{i})$ is determined completely by $\boldsymbol{z}_+(\boldsymbol{i}), o(\boldsymbol{i}), \boldsymbol{a}, \sigma_e^2$ and $\boldsymbol{d}$. The eight-connected neighbors of $o(\boldsymbol{i})$ are indicated by $o(\boldsymbol{i}+\boldsymbol{q}_k)$, with $k = 1, \ldots, 8$.

Drawing samples from (B.16) with the Gibbs sampler is very easy. It involves evaluating (B.16) at a specific site $\boldsymbol{i}$ for $o(\boldsymbol{i}) = 0$ and for $o(\boldsymbol{i}) = 1$, while keeping the other values for the direction mask and the $\hat{y}(\boldsymbol{i})$ fixed. The results are assigned to c_1 and c_2, respectively. Next, a value for $o(\boldsymbol{i})$ (and thereby the corresponding $\hat{y}(\boldsymbol{i})$) is chosen at random, with a probability $c_1/(c_1+c_2)$ that $o(\boldsymbol{i}) = 0$ and with a probability $c_2/(c_1+c_2)$ that $o(\boldsymbol{i}) = 1$. A single update of an image region consists of applying the Gibbs sampler to each site in that region in turn, following, for instance, a checkerboard scanning pattern.

Appendix C

Optimal quantizers for encoding noisy image sequences

This appendix shows that minimizing the error variance $E[\varepsilon^2(\boldsymbol{i})]$ between input $y(\boldsymbol{i})$ and output $\hat{z}(\boldsymbol{i})$ for a communication system as depicted in Figure 5.9 is equivalent to designing optimal quantizers for the MPEG2 encoder. It is assumed that the channel is error free. Work related to this topic is given in [24,27]. The equation for the optimal quantizers is derived. For ease of notation, spatial indices $\boldsymbol{i}$ are omitted in this appendix.

In the absence of channel errors, the scheme in Figure 5.9 can be simplified to that in Figure C.1 in which the noisy signal is transformed by the DCT, quantized, inverse quantized and inverse transformed. Figure C.2 gives an example of a quantizer with L_k representation levels. The error variance is related to the quantization error in the coded DCT coefficients:

$$
\begin{aligned}
E[\varepsilon^2] &= \sum_{k=1}^{64} E[(Y_k - \hat{Z}_k)^2] \\
&= \sum_{k=1}^{64} E[(Y_k - Q_k[Z_k])^2] \qquad \text{(C.1)} \\
&= \sum_{k=1}^{64} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (Y_k - Q_k[Z_k])^2 \cdot P_{Y_k, Z_k}[Y_k, Z_k] d_{Z_k} d_{Y_k}.
\end{aligned}
$$

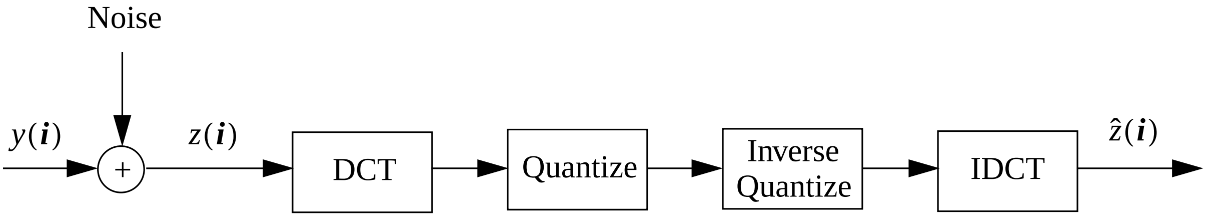


Figure C.1 Simplification MPEG2 encoding/decoding over a noise free channel.

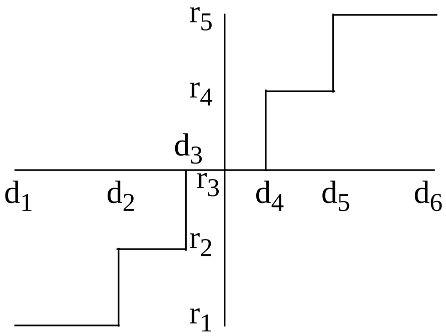


Figure C.2 Example of quantizer with representation levels r_1 to r_5 and decisions levels d_1 to d_6. Note that d_1 and d_6 lie at plus and minus infinity.

Here Y_k, $\hat{Z}_k$ and Z_k, with $k = 1, \ldots, 64$, indicate DCT coefficients with coefficient number k obtained from 8×8 image blocks. The quantizer for DCT coefficient k is indicated by $Q_k[\cdot]$. The joint probability distribution $P_{Y_k, Z_k}[Y_k, Z_k]$ is given by:

$$\begin{aligned} P_{Y_k, Z_k}[Y_k, Z_k] &= P_{Z_k}[Z_k|Y_k] \cdot P_{Y_k}[Y_k] \\ &= P_N[Y_k - Z_k] \cdot P_{Y_k}[Y_k], \end{aligned} \tag{C.2}$$

where $P_N[\cdot]$ is the distribution of the additive noise.

Because the MPEG2 coding standard defines the representation levels of the inverse quantizer in the decoder, the only free parameters in the chain from input to output are the decision levels of the quantizers in the encoder. Minimizing (C.1) is therefore equivalent to selecting optimal decision levels $d_{k,m}$ in the quantizer, given the representation levels $r_{k,l}$, with $l = 1, \ldots, L_k$ and $m = 1, \ldots, L_k + 1$.

Without loss of generality, let $d_{k,1} = -\infty$ and $d_{k,L+1} = \infty$. Equation (C.1) can then be broken down into L partial integrals over the L decision intervals:

$$\begin{aligned} E[\varepsilon^2] &= \sum_{k=1}^{64} \int_{-\infty}^{\infty} \sum_{l=1}^{L_k} \int_{d_l}^{d_{l+1}} (Y_k - Q_k[Z_k])^2 \cdot P_N[Y_k - Z_k] \cdot P_{Y_k}[Y_k] d_{Z_k} d_{Y_k} \\ &= \sum_{k=1}^{64} \sum_{l=1}^{L_k} \int_{-\infty}^{\infty} (Y_k - r_{k,l})^2 \cdot P_{Y_k}[Y_k] \cdot \int_{d_l}^{d_{l+1}} P_N[Y_k - Z_k] d_{Z_k} d_{Y_k}. \end{aligned} \tag{C.3}$$

Equation (C.3) is always positive. Hence, it is minimized by minimizing each of its 64 terms, i.e., by selecting the optimal decision levels given the representation levels for all the individual quantizers Q_k. The optimal decision levels are obtained by setting the derivatives with respect to $r_{k,l}$ to zero. This yields the following decision levels $d_{k,m}$, with $2 \le m \le L_k$:

$$\int_{-\infty}^{\infty} (((r_{k,m-1} - Y_k))^2 - (r_{k,m} - Y_k)^2) \cdot P_{Y_k}[Y_k] \cdot P_N[Y_k - d_{k,m}] dY_k = 0. \tag{C.4}$$

The optimal quantizers are now defined. Some concluding remarks can now be made. First, note that in an MPEG2 encoder, the input signal $y(\boldsymbol{i})$ in Figure C.1 can be either a true image or an image representing prediction errors, depending on whether an I, P, or B frame is coded. The statistics for these images vary, and therefore different quantizers need to be computed for each situation. Second, depending on the amount of bits that are available, an MPEG2 encoder selects a quantizer with a certain number of quantization levels. To get minimum error variance, multiple optimal quantizers have to be computed to accommodate this freedom of the encoder.

Bibliography

[1] E. Abreu, M. Lightstone, S.K. Mitra, and K. Arakawa, “A New Efficient Approach for the Removal of Impulse Noise from Highly Corrupted Images”, IEEE Trans. on Image Processing, Vol. 5, No. 6, pp. 1012-1025, 1996.

[2] E.H. Adelson, C.H. Anderson, J.R. Bergen, P.J. Burt, and J. M. Ogden, “Pyramid Methods in Image Processing”, RCA Engineer, Vol. 29, No. 6, pp. 33-41, 1984.

[3] J. Allnat. “Transmitted-picture Assessment”, John Wiley & Sons, 1983.

[4] M.Antonini, T. Gaidon, P. Mathieu, and M. Barlaud, “Wavelet Transform and Image Coding”, in “Wavelets in Image Communication”, M. Barlaud. ed, Elsevier, pp. 65-119, 1994.

[5] G. R. Arce, “Multistage Order Statistic Filters for Image Sequence Processing”, IEEE Trans. on Signal Processing, Vol. 39, No. 5, pp. 1146-1163, 1991.

[6] S. Armstrong, A.C. Kokaram, and P.J.W. Rayner, “Reconstructing Missing Regions in Colour Images using Multichannel Median Models”, Proc. of EUSIPCO 98, Vol. II, pp. 1029-1034, Rhodes, Greece, 1998.

[7] J. Astola, P. Haavisto, and Y. Neuvo, “Vector Median Filters”, Proc. of the IEEE, Vol. 78, No. 4, pp. 678-689, 1990.

[8] M.R. Banham and A.K. Katsaggelos, “Digital Image Restoration”, IEEE Signal Processing Magazine, Vol. 14, No. 2, pp. 24-41, 1997.

[9] M. Barni, F. Bartolini, A. Piva, and F. Rigacci, “Statistical Modelleing of Full Frame DCT coefficients”, Signal Processing IX, Vol. 3, pp. 1513-1516, Rhodes, Greece, 1998.

[10] J. Biemond, L. Looijenga, D.E. Boekee, and R.H.J.N. Plompen. “A pel-recursive Wiener Based displacement estimation Algorithm”, Signal Processing, Vol. 13, No.4, pp. 399-412, 1987

[11] M. Bierling, “Displacement Estimation by Hierarchical Block Matching”, SPIE VCIP, pp. 942-951, Cambridge, U.K., 1988.

[12] F.C. Billingsley, "Noise Considerations in Digital Image Processing Hardware" in "Topics in Applied Physics", Vol. 6, e.d. T.S. Huang, Springer-Verlag, Berlin, 1975.

[13] J.C. Brailean, R.P. Kleihorst, S.N. Efstratiadis, A.K. Katsaggelos, and R.L. Lagendijk, "Noise Reduction Filters for Dynamic Image Sequences: A review", Proc. of the IEEE, Vol 83, no. 9, pp. 1272-1292, 1995.

[14] P.J. Burt and E.H. Adelson, "The Laplacian Pyramid as a Compact Image Code", IEEE Trans. on Comm., Vol. 31, pp. 532-540, 1983.

[15] J. Canny, "A Computational Approach to Edge Detection", IEEE PAMI, Vol. 8, No. 6, 1986.

[16] M.J. Chen, L.G. Chen, and R. Weng, "Error Concealment of Lost Motion Vectors with Overlapped Motion Compensation", IEEE Trans. on Circuits and Systems for Video Technology, Vol. 7, No. 3, June 1997.

[17] J. Davidse, "Analoge Singaalbewerkingstechniek", DUM, Delft, The Netherlands, 1991.

[18] J. Davidse, "Televisie Techniek en Beeldversterking", DUM, Delft, The Netherlands, 1992.'

[19] P. Delogne, L. Cuvelier, B. Maison, B. Van Caillie, and L. Vandendorpe, "Improved Interpolation, Motion Estimation, and Compensation for Interlaced Pictures", IEEE Trans. on Image Processing, Vol. 3, No. 5, 1994.

[20] D.L. Donoho, "De-noising by Soft-Thresholding", IEEE Trans. on Information Theory, Vol 41, No. 3, pp. 613-627, 1995.

[21] D.L. Donoho and I.M. Johnstone, "Ideal Denoising in an Orthonormal Basis hosen from a Library of Bases", Technical Report 461, 1994.

[22] D.L. Donoho and I.M. Johnstone, "Ideal Spatial Adaptation via Wavelet Shrinkage", Biometrika, Vol. 81, pp. 425-455, 1994.

[23] E. Dubois and S. Sabri, "Noise Reduction in Image Sequences using Motion Compensated Temporal Filtering", IEEE Trans. on Communications, no. 32, pp. 826-831, 1984.

[24] Y. Ephraim and R.M. Gray, "A Unified Approach for Encoding Clean and Noisy Sources by Means of Waveform and Autoregressive Model Vector Quantization", IEEE Trans. on Information Theory, Vol. 34, No. 4, 1988.

[25] A. T. Erdem, and C. Eroglu, "The Effect of Image Stabilization on the Performance of the MPEG-2 Video Coding Algorithm", Proc. of VCIP-98, Vol. 1, pp. 272-277, San Jose, California, USA, 1998.

[26] D. Ferrandiere, "Motion Picture Restoration using Morphological Tools", International Symposium on Mathematical Morphology (ISMM), pp. 361-368, Kluwer Academic Press, 1996.

[27] T. Fine, "Optimum Mean-Square Quantization of a Noisy Input", IEEE Trans. on Information Theory, pp. 293-294, 1965.

[28] S. Geman and D. Geman, "Stochastic Relaxation, Gibbs Distribution, and Bayesian Restoration of Images", IEEE Trans. on Pattern Recognition and Machine Intelligence, Vol. 6, pp. 721-741, 1984.

[29] W.B. Goh, M.N. Chong, S. Kalra, and D. Krishnan, "Bi-Directional 3D Auto-Regressive Model Approach to Motion Picture Restoration", Proc. of ICASSP 96, pp. 2277-2280, 1996.

[30] P.C. Goldmark and J.M. Hollywood, "A New Technique for Improving the Sharpness of Television Pictures", JSMPTE, Vol. 57, pp. 382-396, 1951.

[31] G. de Haan, "Motion Estimation and Compensation", Ph.D. Thesis, TU Delft, Delft, The Netherlands, 1992.

[32] P. Haskell and D. MesserSchmitt, "Resynchronization of Motion Compensated Video Affected by ATM Cell Loss", Proc. of ICASSP 1992, Vol. 3, pp. 545-548, San Francisco, USA, 1992.

[33] B.G. Haskell, A. Puri, and A. N. Netravali, "Digital Video: An Introduction to MPEG-2", Chapman & Hall, 1997.

[34] J.G. Hayes, "Numerical Approximation to Functions and Data", Canterbury, England, 1967.

[35] H.P. Hiriyannaiah, G.L. Bilbro, and W.E. Snyder, "Restoration of Piecewise-Constant Images by Mean-Field Annealing", J. Opt. Soc. Amer., pp. 1901-1912, 1989.

[36] M. Holschneider, R. Kronland-Martinet, and J. Morlet, "A real-time algorithm for signal analysis with the help of the wavelet transform", in Wavelets, Time-Frequency Methods and Phase Space, Berlin: springer, IPTI, pp. 286-297, 1989.

[37] IEC/ISO 13818-1 IS "General Coding of Moving Pictures and Associated Audio- Part 1: System".

[38] IEC/ISO 13818-2 IS "General Coding of Moving Pictures and Associated Audio- Part 2: Video".

[39] IEC/ISO 13818-3 IS "General Coding of Moving Pictures and Associated Audio- Part 3: Audio".

[40] IEC/ISO JTC1/SC29/WG11/93-400, "Test Model 5", Test Model Editing Committee, 1993.

[41] ITU-R Rec. BT.500-7, "Methodology for the Subjective Assessment of Quality of Television Pictures", ITU, 1995.

[42] A.K. Jain, "Fundamentals of Digital Image Processing", Prentice Hall, 1989.

[43] R.L. Joshi and T.R. Fisher, 'Comparison of Generalized Gaussian and Laplacian Modeling in DCT Image Coding", IEEE Signal Processing Letters, Vol 2, No. 5, pp. 81-82, 1995.

[44] B. I. Justusson, "Median Filtering: Statistical Properties", in Topics in Applied Physics 43, T.S. Huang ed., pp.161-196, 1981.

[45] S. Kalra, M.N. Chong, and D. Krishnan, "A new Auto Regressive (AR) model-based algorithm for Motion Picture Restoration", Proc. of ICASSP 97, Vol. 4, pp. 2557-2560, Munich, Germany, 1997.

[46] R.P. Kleihorst, "Noise Filtering of Image Sequences", Ph. D. Thesis, TU Delft, Delft, The Netherlands, 1994.

[47] A.C. Kokaram, "Motion Picture Restoration", Springer Verlag, 1998.

[48] A.C. Kokaram, R.D. Morris, W.J. Fitzgerald, and P.J.W. Rayner, "Detection of Missing Data in Image Sequences", IEEE Trans. on Image Processing, pp. 1496-1508, Vol. 4, No. 11, 1995.

[49] A.C. Kokaram, R.D. Morris, W.J. Fitzgerald, and P.J.W. Rayner, "Interpolation of Missing Data in Image Sequences", IEEE Trans. on Image Processing, pp. 1509-1519, Vol. 4, No. 11, 1995.

[50] A.C. Kokaram, P.M.B. van Roosmalen, P.J.W. Rayner, and J. Biemond, "Line Registration of Jittered Video", ICASSP-97, pp. 2553-2556, Munich, Germany, 1997.

[51] J. Konrad and E. Dubois, "Bayesian Estimation of Motion Vector Fields", IEEE Trans. on PAMI, Vol. 14, No. 9, 1992.

[52] R.L. Lagendijk and J. Biemond, "Iterative Identification and Restoration of Images", Kluwer Academic Publishers, 1991.

[53] R.L. Lagendijk and J. Biemond, "Statistische Signaalverwerking", DUM, The Netherlands, 1994.

[54] W. Lam, A.R. Reibman, and B. Liu, "Recovery of Lost or Erroneously Reveived Motion Vectors", Proc. of ICASSP 1993, vol. 5, pp.417-420, Minneapolis, USA, 1993.

[55] A. Leon-Garcia, "Probability and Random Processes for Electrical Engineering", 2nd. Ed., Addison-Wesley, 1994.

[56] S. Mallat, "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation", IEEE Trans. on Pattern Anal. and Mach. Intell., Vol. 11, No 7, 1989.

[57] P. Maragos, "Morphological Signal and Image Processing", in The Digital Signal Processing Handbook, V. Madisettii and D.B. Williams eds., CRC Press, 1998.

[58] J.B. Martens, "Adaptive Contrast Enhancement through Residue-Image Processing", Signal Processing, Vol. 44, pp. 1-18, 1995.

[59] J.H. McClellan, "The design of two-dimensional filters by transformations", Proc. 7th Annual Princeton conference of Sciences and Systems, pp. 247-251, 1973.

[60] R.H. McMann and A.A. Goldberg, "Improved Signal Processing Techniques for Color Television Broadcasting", JSMPTE, Vol. 77, pp. 221-228, 1968.

[61] J. L. Mitchell, W.B. Pennebaker, C. E. Fog, and D. J. LeGall, "MPEG Video Compression Standard", Chapman & Hall, USA, 1996.

[62] R.D. Morris, W.J. Fitzgerald, and A.C. Kokaram, "A Sampling Based Approach to Line Scratch Removal from Motion Picture Frames", Proc. of ICIP-96, vol I, pp. 801-804, Lausanne, Switzerland, IEEE, 1996.

[63] H. Muller-Seelich, W. Plaschzug, P. Schallauer, S. Potzman, and W. Haas, "Digital Restoration of 35mm Film", Proc. of ECMAST 96, Vol. 1, pp. 255-265, Louvain-la-Neuve, Belgium, 1996.

[64] M.J. Nadenau and S.K. Mitra, "Blotch and Scratch Detection in Image Sequences based on Rank Ordered Differences", in Time-Varying Image Processing and Moving Object Recognition, V. Cappelini ed., Elsevier, 1997.

[65] A. Narula and J.S. Lim, "Error Concealment Techniques for an All-Digital High-Definition Television System", Proc. of SPIE, Vol. 2094, pp. 304-315, 1993.

[66] B.K. Natarajan, "Filtering Random Noise from Deterministic Signals via Data Compression", IEEE Trans. on Signal Processing, Vol. 43, No. 11, pp. 2595-2605, 1995.

[67] J.M. Odobez and P. Bouthemy, "Robust Multiresolution Estimation of Parametric Motion Models", Journal of Visual Communications and Image Representation, pp. 348-365, 1995.

[68] J.J.K. Ó Ruanaidh and W.J. Fitzgerald, "Numerical Bayesian Methods Applied to Signal Processing", Springer, USA, 1996.

[69] M.K. özkan, A.T. Erdem, M.I. Sezan, and A.M. Tekalp, "Efficient Multiframe Wiener Restoration of Blurred and Noisy Image Sequences, IEEE Trans. on Image Processing, Vol 1, no 4, pp 453-476, 1992.

[70] M.K. özkan, M.I. Sezan, and M. Tekalp, "Adaptive Motion Compensated Filtering of Noisy Image Sequences", Trans. on Circuits and Systems for Video Technology, Vol 3, No. 4, IEEE, 1993.

[71] J.J. Pearson, D.C. Hines, S. Goldsman, and C.D. Kuglin, "Video Rate Image Correlation Processor", SPIE, Vol. 119, Application of Digital Image processing, 1977.

[72] P.G. Powell and B.E. Bayer, "A method for the Digital Enhancement of Unsharp, Grainy Photographic Images", IEE Int. Conf. on Electronic Image Processing, No. 214, pp. 179-183, 1982.

[73] W.K. Pratt, "Digital Image Processing", John Whiley & Sons, 2nd Ed., 1991.

[74] W.K. Pratt, "Vector Space Formulation of Two-Dimensional Signal Processing Operations", Computer Graphics and Image Processing 4, pp. 1-24, 1975.

[75] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery, "Numerical Recipes in C", 2nd Ed., Cambridge University Press, U.K., 1992.

[76] R.C. Reiniger and J.D. Gibson, "Distributions of the Two-Dimensional DCT coefficients for Images", IEEE Trans. on Communications, Vol. 31, No. 6, pp. 835-839, 1983.

[77] P. Richardson and D. Suter, "Restoration of Historic FIlm for Digital Compression: A Case Study", Proc. of ICIP-95, Vol. II, pp. 49-52, Washington D.C. USA, IEEE, 1995.

[78] R.A. Roberts, C.T. Mullis, "Digital Signal Processing", Addison-Wesley, USA, 1987.

[79] P.M.B. van Roosmalen, J. Biemond, and R.L. Lagendijk, "Restoration and Storage of Film and Video Archive Material", to appear in NATO-ASI Signal Processing for Multimedia.

[80] P.M.B. van Roosmalen, A.C. Kokaram, and J. Biemond, "Fast High Quality Interpolation of Missing Data in Image Sequences using a Controlled Pasting Scheme", Proc. of ICASSP-99, USA, 1999.

[81] P.M.B. van Roosmalen, A.C. Kokaram, and J. Biemond, "Noise Reduction of Image Sequences as PreProcessing for MPEG2 Encoding", Signal Processing IX, pp. 2253-2256, Rhodes, Greece, 1998.

[82] P.M.B. van Roosmalen, R.L. Lagendijk, and J. Biemond, "Improved Blotch Detection by Postprocessing", Proc. of IEEE Signal Processing Symposium SPS'98, pp. 223-226, Leuven, Belgium, 1998.

[83] *P.M.B. van Roosmalen, R.L. Lagendijk, and J. Biemond, "Correction of Intensity Flicker in Old Film Sequences", IEEE Trans. on Circuits and Systems, to appear Dec. 1999.*

[84] *P.M.B. van Roosmalen, R.L. Lagendijk, and J. Biemond, "Flicker Reduction in Old Film Sequences", in: Time-Varying Image Processing and Moving Object Recognition, 4, V. Cappellini, ed., Elsevier Science, pp. 11-18, 1997.*

[85] *P.M.B. van Roosmalen, S.J.P. Westen, R.L. Lagendijk, and J. Biemond, "Noise Reduction for Image Sequences using an Oriented Pyramid Thresholding technique", Proc. of ICIP-96, Vol. I, pp. 375-378, Lausanne, Switzerland, IEEE, 1996.*

[86] *A. Rosenfeld and A. Kak, "Digital Picture Processing", 2nd. Ed., Vol. 1, Academic Press, 1982*

[87] *J. P. Rossi, "Digital Techniques for Reducing Television Noise", JSMPTE, Vol. 87, pp. 134-140, 1978.*

[88] *A. van der Schaaf, "Natural Image Statistics and Visual Processing", Ph. D. Thesis, Rijksuniversiteit Groningen, Groningen, The Netherlands, 1998.*

[89] *K. Sharifi and A. Leon-Garcia, "Estimation of Shape Parameter for Generalized Gaussian Distributions in Subband Decomposition of Video", IEEE Trans. on Circuits and Systems, Vol. 5, No. 1, pp. 52-56, 1995.*

[90] *E.P. Simoncelli and E. H. Adelson, "Noise Removal Via Bayesian Wavelet Coring", Proc. of ICIP-96, Vol. I, pp. 379-382, Lausanne, Switzerland, IEEE, 1996.*

[91] *E. P. Simoncelli, W. T. Freeman, E. H. Adelson, and D. J. Heeger, "Shiftable Multiscale Transforms", IEEE Trans. on Information Theory, Vol. 38, No. 2, pp. 587-909, 1992.*

[92] *G. Strang, "Linear Algebra and its Applications", 3rd Ed., Harcourt Brace Jovanovich, USA, 1988.*

[93] *A. M. Tekalp, "Digital Video Processing", Prentice Hall, USA, 1995.*

[94] *C.W. Therrien, "Discrete Random Signals and Statistical Signal Processing", Prentice Hall, USA, 1992.*

[95] *P.P. Vaidyanathan, "Multirate systems and filter banks", Prentice Hall, pp. 337-338, 1992.*

[96] *R.N.J. Veldhuis, "Adaptive Restoration of Unknown Samples in Discrete-Time Signals and Digital Images", Ph. D. Thesis, Katholieke Universiteit Nijmegen, Nijmegen, The Netherlands, 1988.*

[97] *M. Vetterli and J. Kovacevic, "Wavelets and Subband Coding", Prentice Hall, USA, 1995.*

[98] *T. Vlachos and G. Thomas, "Motion Estimation for the Correction of Twin-Lens Telecine Flicker", Proc. of ICIP-96, Vol. I, pp. 109-112, Lausanne, Switzerland, IEEE, 1996.*

[99] *S. Voran and S. Wolf, "An Objective Technique for Assessing Video Impairments", IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, 1993.*

[100] *B.A. Wandell, "Foundations of Vision", Sinauer Associates, USA, 1995.*

[101] *J.Y.A. Wang and E. Adelson, "Representing Moving Images with Layers", IEEE Trans. on Image Processing, Vol. 3, pp. 625-638.*

[102] *S.J.P. Westen, R.L. Lagendijk, and J. Biemond, "Perceptual Image Quality based on a Multiple Channel HVS Model", Proc. of ICASSP-95, pp. 2351-2354, USA, 1995.*

[103] *J.K. Wolf and J. Ziv, "Transmission of Noisy Information to a Noisy Receiver With Minimum Distortion", IEEE Trans. on Information Theory, Vol. 16, No. 4, pp. 406-411, 1970.*

[104] *E. Wong, "Two-Dimensional random fields and representation of Images", SIAM, Journal of Applied Mathematics, Vol. 16, No. 4, pp. 756-770, 1968.*

[105] *J. Woods, "Two-Dimensional discrete Markov Random Fields", IEEE Trans. on Information Theory, Vol. 18, No 2, pp. 232-240, 1972.*

Samenvatting

Restauratie van Gearchiveerde Film en Video

Wereldwijd liggen unieke verzamelingen beeldmateriaal opgeslagen in grote archieven. Vele historische, artistieke en culturele ontwikkelingen van de 20^e eeuw zijn hierin vastgelegd. Door verouderingsprocessen zijn veel beelddragers echter sterk aangetast. Om te voorkomen dat visueel materiaal van belangrijke momenten in onze geschiedenis verloren gaat is conservering en restauratie een eerste vereiste. Het veiligstellen van ons historisch en cultureel verleden is niet alleen van belang voor de wetenschap. Door het gebruik van digitale videostandaarden zullen in de nabije toekomst nog meer zenders te ontvangen zijn in de woonkamer. Om de zendtijd te vullen zijn programma's nodig. Deze kunnen uiteraard verkregen worden door nieuwe programma's te produceren. Een goedkoop alternatief is het hergebruik van grote kollekties films, series, documentaires en spelprogramma's die zich momenteel in de archieven bevinden. Hierbij dient wel een kanttekening geplaatst te worden. De moderne kijker zal alleen dan oude programma's accepteren indien de visuele- en audiokwaliteit daarvan voldoen aan de eisen van deze tijd.

Vanwege de enorme hoeveelheden opgeslagen film- en videomateriaal, maar echter ook vanwege economische motieven, is het noodzakelijk dat beeldrestauratie wordt uitgevoerd door middel van een automatisch beeldrestauratiesysteem. De nadruk dient gelegd te worden op het woord *automatisch*. Dit omdat handmatige beeldrestauratie een tijdrovende en eentonige aangelegenheid is. In 1995 werd het AURORA-projekt gestart met subsidie van het Europese ACTS programma. AURORA staat voor AUtomated Restoration of ORiginal video and film Archives. Het doel van dit 3 jaar durende project was om een *real-time* systeem te ontwikkelen voor restauratie van oude video- en filmbeelden. Dit systeem moest in staat zijn om grote hoeveelheden materiaal te verwerken met een minimum aan menselijke interaktie. De toenmalige apparatuur vereiste veel menselijke interaktie en kon veel voorkomende soorten artefakten (degradaties van beelden) niet volautomatisch restaureren.

De Technische Universiteit Delft nam deel aan het AURORA projekt. Dit proefschrift beschrijft de onderzoeksresultaten die in Delft behaald werden in het kader van dit projekt. De volgende verstoringen werden in Delft onderzocht: knipperen van de helderheid in beelden, vlekken en ruis. Het knipperen van de helderheid is een veel voorkomend artefakt in oude zwart-wit films dat wordt ervaren als tegennatuurlijke temporele fluctuaties in beeldintensiteit. Dit proefschrift beschrijft een orginele en effectieve methode om het knipperen in helderheid te corrigeren. Deze methode is gebaseerd op het gelijktrekken van het lokale gemiddelde en de lokale variantie van de beeldintensiteit in opeenvolgende beelden.

Vlekken zijn typisch film-gerelateerde artefakten. Ze worden veroorzaakt doordat de gelatine van de film loslaat en doordat vuil zich aan de film hecht. Bestaande methoden voor het detecteren van vlekken maken veel fouten in de zin dat ze voor te veel beeldelementen ten onrechte aangeven dat deze deel uitmaken van een vlek (loos alarm). Het gevolg is dat er in de gecorrigeerde sequentie fouten kunnen ontstaan die, visueel gezien, nog storender zijn dan de oorspronkelijke vlekken. Dit komt omdat de gebruikte interpolatietechnieken om beelden te corrigeren ook fouten maken. Dit proefschrift beschrijft technieken om de detectie van vlekken te verbeteren. Deze technieken houden rekening met de invloed van ruis op de detector en buiten de spatiële coherentie die eigen is aan vlekken uit. Bovendien is er een nieuwe, op beeldmodellen gebaseerde methode voor het corrigeren van vlekken ontwikkeld die sneller en meer robust is dan bestaande methoden.

Coring is een bekende techniek om ruis te verwijderen uit beelden. Eerst wordt het geobserveerde beeldsignaal naar een frequentiedomein getransformeerd. Het getransformeerde signaal wordt dan aangepast volgens de zogenaamde coringkarakteristiek. Het eindresultaat (het beeld waaruit de ruis is verwijderd) wordt verkregen door de inverse transformatie toe te passen op de aangepaste data. In dit proefschrift wordt een raamwerk ontwikkeld om deze techniek te kunnen toepassen op video- en filmsequenties. Dit raamwerk is gebaseerd op drie-dimensionale beeldtransformaties. Deze beeldtransformaties staan toe dat informatie in de temporele dimensie uitgebuit kan worden ten behoeve van het ruisverwijderingsproces. Dit laatste is niet mogelijk in de situatie waarin de beelden afzonderlijk van elkaar worden bewerkt. Dit proefschrift laat tevens zien dat coring binnen het MPEG2 codeerschema geplaatst kan worden zonder dat dit al te veel aan complexiteit toevoegt. MPEG2 wordt dan een systeem dat simultaan ruis verwijdert en beelden comprimeert. Het aangepaste codeerschema levert een significante verhoging in de kwaliteit van gecodeerde, ruisige beeldsequenties.

Beeldrestauratie verhoogt niet alleen de perceptuele kwaliteit van video- en filmbeelden, maar het vergroot ook de beeldcoderingsefficiëntie. Dit betekent dat bij een gegeven vast aantal bits beelden met hogere kwaliteit gecodeerd kunnen worden. Andersom kan met minder bits dezelfde kwaliteit behaald worden. Dit laatste is vooral van belang in situaties waarin beelddata digitaal wordt uitgezonden en opgeslagen. In deze omstandigheden lopen de kosten op met het benodigde aantal bits. Dit proefschrift onderzoekt en evalueert de invloed van artefacten op de coderingsefficiëntie. Aangetoond wordt dat flink op bandbreedte bespaard kan worden zonder kwaliteitsverlies.

Acknowledgements

A thesis is never the result of the work of a single person. I am indebted to many people who contributed either directly or indirectly to this thesis. Listing everyone who made the contributions would undoubtedly fill quite a few pages and so I will restrict myself to a number of people I would like to give special mention.

First of all, I would like to thank my parents for their support and encouragement over and for providing me with the means to getting this far. I would also like to thank Conny for being so patient ☺

I would like to thank Jan Biemond for inviting me to take part in the Aurora project, for his good advice and support over the years, and for being pleasant travel company. Furthermore, I am also grateful to my roommate Inald Lagendijk for the technical discussions we had and for the enormous amount of time and interest he took in carefully reading and commenting on my thesis.

The results in this thesis are also due to the interactions that took place within the Aurora project. The partners of the Aurora project gave birth to many interesting discussions. I would especially like to mention Anil Kokaram, Jean-Hugues Chenot, Louis Laborelli, John Drewery, Jim Easterbrook, Colin Smith, and Tomaso Erseghe for opening up my eyes towards concepts of parallel processing, keeping things nice and simple, internet economics, Gregorian chanting, bicycle hubs, the court of law, and Indian food.

A thesis is not only about knowledge. It is also about communication and language. Therefore, I would like to thank my mother (again) for editing the draft version of this thesis. Nowadays, image processing is all about computing and writing a thesis is all about desk-top publishing. To come to a thesis that deals with image processing requires a good computer infrastructure

that is well managed. I give special thanks to Ben van den Boom for keeping my computer and the network up and running in a smooth fashion and for readily solving problems as they arose.

Furthermore, I would like to mention that I enjoyed the lunches very much with (in no particular order) Erik “There is no conspiracy”, Andre “Let’s not talk about the piano”, Alan “Schnitzel essen?”, Gerhard “I have a problem...(sigh)” and Isabel “All the boards in the cupboard are mine!”

Finally, I would like to thank all colleagues at the information and communication theory group that I have not mentioned explicitly, such as Erik Vullings and John Schavemaker, for making my years in Delft very pleasant ones.

Curriculum Vitae

Peter Michael Bruce van Roosmalen was born in Maastricht, the Netherlands, on July 16, 1970. In 1987, he received his HAVO diploma (general secondary education) from the Henric van Veldeke College in Maastricht. Two years later he received his Atheneum diploma from this college. Next, he studied Electrical Engineering at the Delft University of Technology. He worked at Philips Medical Systems as a trainee for four months in 1993. He carried out his M. Sc. project at the Information Theory Group (currently the Information and Communication Theory Group), Delft University. The topic of the project was 3D modeling of dolphin CT scans. He received his M. Sc. in September 1994.

After an extensive holiday, he became a member of the technical staff of the Information Theory Group in 1995. There he designed and implemented an operating system for recording and playing back image sequences in stereo in real-time. In September of that same year, he joined the AURORA project and became a Ph. D. student. After writing his dissertation, he did a pilot study in favor of a sequel to the AURORA project at INA in Paris, France.