Noise Filtering of Image Sequences

Ruisfilteren van Beeldsequenties

Noise Filtering of Image Sequences

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 van Dekanen aangewezen,
op maandag 10 oktober 1994 te 10:30 uur
door

Richard Petrus KLEIHORST,

elektrotechnisch ingenieur,

geboren te Schiedam.

Dit proefschrift is goedgekeurd door de promotor:
Prof.dr.ir. J. Biemond,
en de toegevoegd promotor: Dr.ir. R. L. Lagendijk.

Promotiecommissie:

Rector Magnificus
Prof.dr.ir. J. Biemond (promotor)
Dr.ir. R.L. Lagendijk (toegevoegd promotor)
Prof.dr. A.K. Katsaggelos
Prof.dr.ir. A.W. Heemink
Prof.ir. G. Honderd
Prof.dr.ir. R.H.J.M. Otten
Dr.ir. G. de Haan

CIP-DATA KONINKLIJKE BIBLIOTHEEK, DEN HAAG

Kleihorst, Richard Petrus

Noise filtering of image sequences / Richard P.
Kleihorst. - Delft : Technische Universiteit Delft,
Faculteit der Elektrotechniek. - Ill.
Thesis Technische Universiteit Delft. - With ref. - With
summary in Dutch.
ISBN 90-5326-018-8
NUGI 832
Subject headings: Digital signal processing, / video
processing, / order statistics.

Preface

This thesis reports on the results of the research in the field of noise-filtering of image sequences as carried out in the Information Theory Group at Delft University of Technology, the Netherlands, from 1990 to 1994. The work started out as a Chartered Designer's project which was successfully completed in January 1992. After that, it was continued as a Ph.D. research project.

Investigations into noise-filtering of image sequences seriously took off in the early 80s. The Information Theory Group has been active in this field of research since the second half of the 80s. Part of the work has been carried out in collaboration with the Digital Signal and Image Processing Laboratory at the Technological Institute of Northwestern University, Robert R. McCormick School of Engineering and Applied Science, Illinois, USA. The work has resulted in a number of (joint) publications.

Contents

Summary

Image sequences are digital recordings of time-varying 3D phenomena. Often, the sequences are corrupted by noise, introduced by the image sensor and therefore inherently present in the imaging process. This noise can be reduced to improve the visual appreciation and the results of subsequent processing tasks. The reduction is performed by noise-filtering algorithms.

The 3D image-sequence signals have two spatial coordinates, indexing the spatial plane, and one coordinate in time, indexing the temporal direction. In video signals, the signals in the spatial plane and the signals in the temporal direction have different properties. As these properties concern the predictability and homogeneity of the signal, they are of importance for noise filtering. In the spatial plane the signal comprises image data, and is non-stationary. In the temporal direction two situations can occur. First, in an inactive part of the sequence the temporal signals are quite homogeneous and therefore particularly suited for noise filtering. Second, in an active part of the sequence moving objects and texture are passed, making the temporal signal highly non-stationary. At most, parts of this signal have a homogeneous character.

As non-stationarities in temporal signals are often due to object motion, a possible avoidance is achieved by compensating the sequence for motion. This comprises estimating the motion and indexing the signal along the motion trajectory. The resulting "motion-compensated" noise filtering is the subject of Chapter 3. In practice, motion-compensated noise filtering benefits from the increased homogeneity in the temporal direction. However, motion estimation is usually not perfect, due to the noise in the observation and the incompleteness of the motion models used. Because of this imperfectness, it can be used as a sensible first strategy, but (also because not all non-stationarities are caused by motion) non-stationarities are still encountered along the motion-compensated temporal trajectory. To decrease the influence of noise on the motion estimate we have used a recursive-search block matcher, extended with a novel noise-insensitive criterion function based on third-order statistics. With this modification, the motion estimation is reliable even with severe noise.

The lack of a definite homogeneous signal path in the image sequence, both spatial and temporal, dictates the use of signal-adaptive filtering algorithms. Several adaptive filtering algorithms have been proposed in literature, such as filters that switch-off the filtering action in non-stationary regions, order-statistic filters, and filtering algorithms that perform a segmentation step prior to filtering. Both temporal and spatio-temporal data windows are used for filtering. Overall, filters with a spatio-temporal data window achieve a higher noise suppression because of the increased amount of data. However, this higher suppression is achieved with more computational effort. Chapter 2 of this thesis gives an overview of noise-filtering algorithms.

In Chapter 4 we propose a novel adaptive spatio-temporal filtering approach based on ideas from non-stationary time-series processing, namely trend-removal and normalization prior to filtering. This algorithm uses the conception that a non-stationary observed signal can be decomposed into two parts, namely, a non-stationary part, consisting of the trend and the scale factor for normalization, and a part consisting of the normalized signal. The noise, assumed to be stationary, is entirely mapped into the normalized signal. This means that only the normalized signal has to be filtered, which is a relatively easy task as it can be reasonably performed by simple noise-smoothing filters.

As the trend corresponds to local mean and the scale factor for normalization corresponds to local deviation, estimates of these local statistics are necessary for the signal decomposition. We develop adaptive estimators that use statistical properties of ordered observations. The ordered observations can be matched, using linear regression, to ordered values from a given normalized distribution. The optimal regression parameters are estimates of the local mean and deviation. By including temporal recursion, the normalized distribution is adapted to slowly changing statistics within the image sequence.

A popular adaptive noise filter is the Linear Local Minimum Mean Square Error (LLMMSE) filter that uses estimates of the local statistics to adapt the transfer function to local signal properties. This filter is a member of the class of switching filters which switch-off filtering in the case of non-stationary situations to avoid smoothing (moving) objects. In the case of homogeneous situations, they have full noise suppression power. In Chapter 5 we propose the novel combination of the LLMMSE filter with order statistics based estimators for estimating the filter's parameters, namely the local mean and deviation.

In the LLMMSE filter, the estimates of the local statistics have a direct influence on the filter output. This demands a high estimation accuracy when outlying observations, non-stationarities, are present in the data. To guarantee this we consider robust estimation techniques that detect and remove outlying observations from the estimation process. We have proposed two techniques. First, an iterative technique involving robust regression, where detection and estimation are performed simultaneously. Second, a separate a-priori test for outlying observations, where the detected outliers are ignored in the estimation process. This test is efficient because it is performed on ranges instead of on individual observations. In addition, the decision efficiency is consistent for all data-window sizes.

Usually, the observation noise is assumed to be additive and signal independent which is valid in many practical situations. However, in some practical situations, the observation noise is signal dependent. This means that filters that are based on the independence assumption will give sub-optimal results, or even fail. We have devised new filtering algorithms in Chapter 6 for two applications where signal dependent noise is encountered. The first application is clinical X-ray image sequences, where as a result of lowering the radiation dosage, the images are quantum limited. The quantum-limited imaging process inherently suffers from signal-dependent noise with a Poisson-shape probability density function. Using the properties of order statistics we propose a dedicated estimator in combination with outlier rejection. The second application is gamma-corrected video signals, where because of the non-linear gamma-correction stage, the observation noise has signal-dependent statistics. For this application we derive an order-statistic filter with fixed coefficients. This new filter uses higher-order order statistics for optimal estimation of the gamma-corrected original intensity. In combination with the separate test for ranges, outliers in the data are detected and removed prior to filtering.

Chapter 1

Introduction to image sequence noise filtering

Image sequences are consecutive digital recordings of a time-varying 3D scene [1]. Image sequences are dealt with in a number of applications, for instance in broadcast, video-phone, tele-conferencing systems [2], satellite observation or surveillance systems, and clinical radiology [3, 4]. Often, image sequences are corrupted by some amount of noise. This can be caused by imperfections of the scanner, or even be inherent to the image-formation process itself. Usually, the various corrupting noise sources are lumped to one additive entity. In this way, the noisy, observed image sequence $g(i,j,k)$ can be expressed by the following *observation model*:

$$g(i,j,k) = f(i,j,k) + n(i,j,k), \tag{1.1}$$

where $f(i,j,k)$ denotes the original signal and $n(i,j,k)$ the noise. The indices i, j reflect the vertical and horizontal, or *spatial* indices, and k is the *temporal* index or frame number (see Figure 1.1).

In most situations, we will assume that the additive noise is zero-mean, white, independent of $f(i,j,k)$ and Gaussian distributed with constant variance. We will restrict ourselves to sequences containing gray-value images that are available in a non-interlaced or progressive form [5].

The amount of corruption by the noise is expressed by the Signal-to-Noise Ratio (SNR) defined as:

$$\mathrm{SNR} = 10 \log_{10} \frac{\text{(original signal variance)}}{\text{(noise variance)}} \quad \text{(dB)}. \tag{1.2}$$

Because of amplitude quantization, the SNR is limited to about 40 to 50dB for practical data quantized in 256 levels.

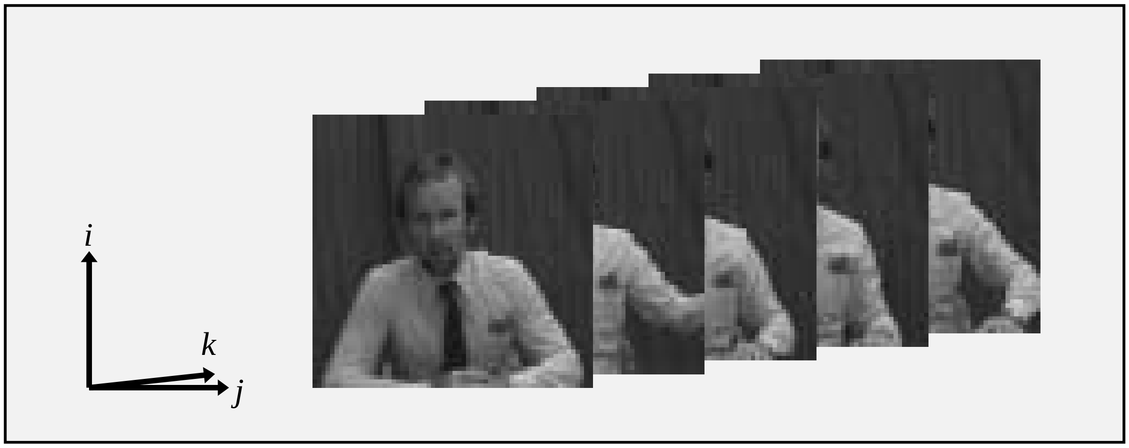


Figure 1.1: Image sequences are consecutive digital recordings of a time-varying scene

To improve visual appreciation and to facilitate subsequent coding or analysis, the noise can be reduced by noise filtering [6, 7, 8]. This filtering is the topic of this thesis.

1.1 Image sequence noise filtering

Of importance to noise filtering is the fact that in image sequences two kinds of approaches to the same signal with different properties can be distinguished. These sub-signals are the spatial and temporal signals and their properties influence the choices for filter support and filter structure.

The spatial properties of images (k fixed) are widely known from image processing literature [9]. It was found that typical spatial information is non-stationary [10, 11, 12]. At best, the spatial signal consists of multiple wide-sense stationary regions. For this reason it is required to use adaptive processing when dealing with spatial signals

Temporal signals are obtained if we consider $f(i, j, k)$ for a fixed spatial position (i, j), indicated by $f(k)$. Two situations can be distinguished with respect to the properties of $f(k)$: $f(k)$ can originate from an inactive part or from an active part of the sequence.

In an inactive part of the sequence, where no object motion or scene changes occur, the temporal signals are wide-sense stationary and highly correlated. Figure 1.2 gives an example of such a signal. Inactive parts of the sequence are encountered, for instance, in a static background region.

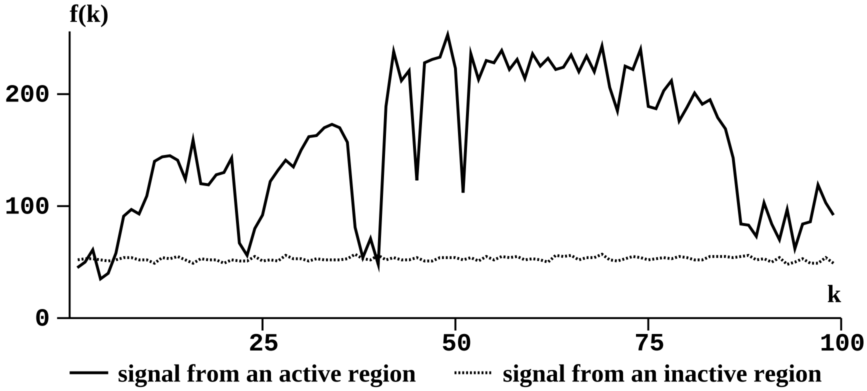


Figure 1.2: The differences between signals from active and inactive regions are clearly recognizable

In an active part of the sequence, the temporal signal is non-stationary because moving objects introduce intensity transitions or temporal edges. At most, the temporal signal is wide-sense stationary in a short time window. Figure 1.2 also gives an example of a temporal signal from an active part of a sequence.

From the above we can conclude that the *main problem* connected with noise filtering of image sequences is how to deal with spatio-temporal non-stationary signals. One way of dealing with temporal edges caused by object motion is to filter the sequence along the motion trajectory. This is called motion compensation and involves motion estimation. However, motion compensation is never perfect in practical sequences [8, 13, 14]. Therefore, this thesis is directed towards the design of *adaptive* spatio-temporal filter structures that can be applied with or without motion compensation.

Most noise filters estimate the original intensity $f(i,j,k)$ from the observations $g(i,j,k)$ within a spatio-temporal window that is associated with the filter support. This support has a finite spatio-temporal extent and contains noisy observations $g(i,j,k)$ and/or previous filtering results $\hat{f}(i,j,k)$ in case of recursive filters. We can distinguish the filtering methods by their supports.

Purely *spatial methods* where the images are filtered separately have been investigated extensively in the field of image filtering [9, 12, 15]. A drawback of this approach is that the strong temporal correlation in image sequences is not used. In addition, temporally inconsistent results may be obtained, causing annoying artifacts. An advantage is that additional motion blurring is avoided. Although purely spatial methods for image sequence filtering will be often used in hardware implementations, they are hardly reported in scientific literature. We will not deal with spatial methods in this thesis. The emphasis will be on filtering approaches that combine spatial and temporal data.

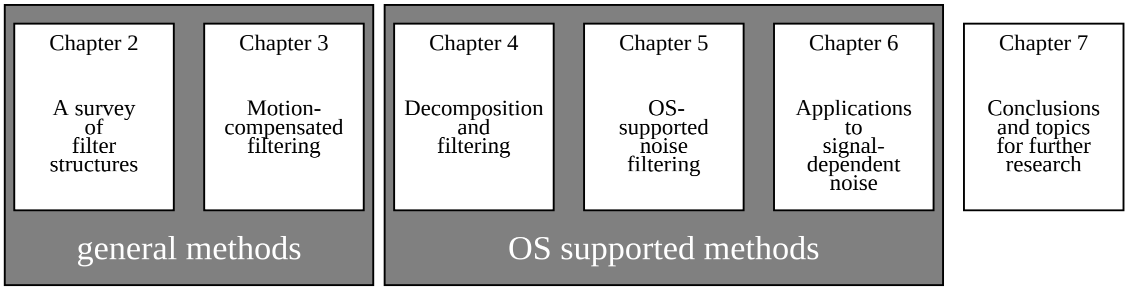

Chapter 2
A survey of filter structures
Chapter 3
Motion-compensated filtering
Chapter 4
Decomposition and filtering
Chapter 5
OS-supported noise filtering
Chapter 6
Applications to signal-dependent noise
Chapter 7
Conclusions and topics for further research
general methods
OS supported methods

FIGURE 1.3: A SCHEMATIC OVERVIEW OF THE CONTENTS OF THIS THESIS

1.2 Outline of the thesis

This thesis is roughly organized into two groups of chapters (see Figure 1.3). In the first group, Chapters 2 and 3, we will revisit image sequence filtering methods and classify them according to their structure. In the second group, Chapters 4, 5 and 6 we present a new direction in image-sequence filtering: using filter techniques involving order statistics.

In Chapter 2 filter *structures* are investigated. This chapter contains a classification and a concise overview of most of the filtering methods that were published. It shows the importance of filter structure for the final filtering result. To stress the importance of the filter structure, we will not yet consider motion compensation. In this light we discuss the use of classical filter techniques, weighed-averaging filters, order-statistic filters, (recursive) switching filters, decomposition methods and Bayesian approaches. This chapter concludes with an experimental evaluation of some representative filters.

Chapter 3 deals with the use of *motion-compensation* in image sequence filtering. We discuss the technique of compensating temporal signals and some of the popular motion-estimation algorithms that are exploited. As motion estimation has to be performed on the noisy observations, various ways to decrease the noise sensitivity of motion estimators are discussed. Here, we also discuss noise-filtering techniques that simultaneously estimate motion and original intensity from the noisy observations. Again, an experimental investigation is presented of several motion-compensated filtering methods.

Chapter 4 investigates the use of trend removal and normalization to *decompose* the observation into a non-stationary part and a wide-sense stationary part. It appears that only the wide-sense stationary part has to be filtered, which requires only relatively simple filters. For trend removal and normalization, estimates of the local statistics (local mean and deviance) are necessary. Adaptive estimators using ordered observations, also known as Order-Statistics (OS) estimators, are derived for this purpose. Chapter 4 is concluded with an experimental evaluation of this novel decomposition method.

In Chapter 5 we address the *combination* of the OS estimators with the popular Linear Local Minimum Mean Square Error (LLMMSE) filter which uses estimates of the local statistics. Usually, this filter employs simple box-average estimators for local mean and deviance which cause sub-optimal results in the case of non-stationary situations. We increase the performance of the LLMMSE filter by supplying accurate estimates of the statistics even in the case of severely non-stationary observations. To this end, we improve the OS estimators by selecting homogeneous parts of the spatio-temporal estimator support and using the appropriate part for estimation. Employing the properties of ordered statistics, we have devised two methods to perform this selection. First, an iterative method where the selection and estimation are performed simultaneously. Second, a method where the selection is performed in a preprocessing step. Chapter 5 closes with an experimental evaluation of the OS-supported LLMMSE filter.

In Chapter 6 we consider two *applications* of the techniques derived in Chapters 4 and 5 to situations where the noise is signal-dependent. As a first application we look at quantum-limited image sequences that arise in clinical radiology. In these systems, the image intensity is reflected by a (low) number of photons per pixel. Inherent to this image formation process, quantum noise is introduced which has statistics that are related to the intensity of the image source. We propose the use of a dedicated OS estimator for these sequences. The second application is the reduction of noise originating from the electronics in video cameras. The gamma-correction stage within the camera causes the observed noise to be signal-dependent. Novel robust OS filters, employing higher-order ordered statistics are proposed. Both methods are experimentally evaluated using synthetically corrupted data and practical data.

Chapter 7 summarizes the conclusions reached in the thesis. Also, some suggestions for future research in the area of image-sequence noise filtering are given.

Appendix A considers the stochastic properties of order statistics. It is included to support the material of Chapters 4, 5 and 6, which deals mainly with applications of order-statistic estimators to image-sequence filtering.

Chapter 2

A survey of filter structures

2.1 Introduction

In this chapter, we consider the basic filter structures that have been used within the field of research on image-sequence noise filtering during the past decade. The main objective is to analyze the different "flows of interests" and categorize them accordingly. We look at and compare the most well-known methods including some proposed by the author which will be investigated more thoroughly in later chapters. The material in this chapter is presented in literature in [16, 17].

The signals of image sequences are highly non-stationary because of spatial and temporal edges. Because the temporal edges are caused by object motion, motion compensation is an obvious part of the filtering scheme. However, motion estimation and compensation are far from perfect because of observation noise and incompleteness of the motion model [14, 18, 19]. In addition, non-stationarities are also caused by changes of scene and lighting condition. As a result, motion-compensated temporal signals can still contain non-stationarities. The spatial signals, being image data, are in general also non-stationary. This leads to the conclusion that any filter, whether using motion compensation or not, will have to manage non-stationary signals. The way in which the filter performs this action, is reflected in the filter structure. For this reason, this chapter focusses on filter structures and in particular on how signal adaptation is included in this structure.

The following classes of filters are described: Section 2.2 deals with filter structures using weighed averaging. In most examples, the weights of these Finite Impulse Response (FIR) filters are adapted to the signal. In Section 2.3 weighed-averaging filters that order the data according to magnitude prior to filtering are considered. These order-statistic filters are edge preserving and are therefore very suited for image (sequence) processing. In Section 2.4 filters are described that adapt by switching off filtering if non-stationarities are detected. They are mostly Infinite Impulse Response (IIR) filters. In Section 2.5 we describe a method that uses trend removal and normalization in order to create a wide-sense stationary signal that can be filtered by a simple filter. In Section 2.6 we deal with Bayesian estimation methods. Here, a local criterion function is optimized by an iterative procedure.

In Section 2.7 the performances of several of the more popular noise filters are compared by noise filtering of synthetically degraded image sequences.

2.2 Weighed averaging

A weighed-averaging filter well known for its use in noise filtering of time-series and images, is the Wiener filter [9, 20, 21]. As a logical extension, this filter was used for noise filtering and de-blurring of image sequences by Özkan *et al.* [22] and Erdem *et al.* [23]. The Wiener filter for noise filtering is given by:

$$\hat{\boldsymbol{f}} = \boldsymbol{R}_f(\boldsymbol{R}_f + \boldsymbol{R}_n)^{-1}\boldsymbol{g}, \tag{2.1}$$

where $\hat{\boldsymbol{f}}$ and $\boldsymbol{g}$ are the estimated and observed image sequences, respectively, written down as lexicographically ordered vectors. $\boldsymbol{R}_f$ and $\boldsymbol{R}_n$ are the correlation matrices of the original signal and noise, respectively.

There are four major disadvantages involved in this approach. First, the requirement that the 3D auto-correlation function for the original sequence has to be known a-priori. Second, the 3D stationarity assumption. This assumption is detrimental to the performance of the 3D Wiener filter. A third disadvantage is the huge number of calculations, which are performed in the frequency domain. This also severely limits the number of frames used in this off-line method. A fourth disadvantage is that in practical motion-compensated schemes only compensation for global translational motion can be incorporated, using the shifting properties of the Fourier transform [23].

The Wiener filter from (2.1) usually involves a global image sequence operation, i.e. all data is required to obtain a single element of the estimated sequence $\hat{\boldsymbol{f}}$. Far more attractive are filters that perform local weighing in a restricted filter support:

$$\hat{f}(i,j,k) = \sum_{p,q,l \in S} w_{p,q,l}(i,j,k) g(i-p, j-q, k-l), \tag{2.2}$$

where S is the filter support and $w_{p,q,l}(i,j,k)$ are the filter weights. The sum of the filter weights has to be unity in order to have an estimate which is free of bias. In a spatio-temporal filter, S is a three-dimensional support. A typical support is:

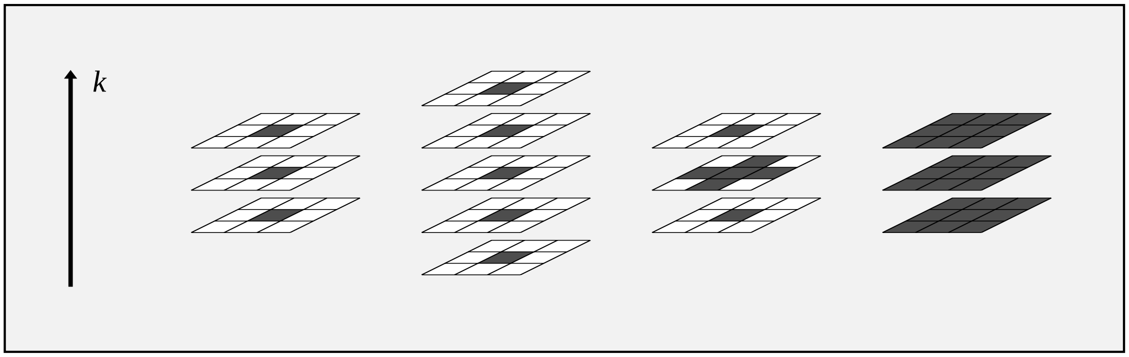


FIGURE 2.1: SOME EXAMPLES OF SPATIO-TEMPORAL FILTER SUPPORTS. THE DARK PIXELS IN THESE EXAMPLES ARE ACTIVE IN THE FILTER SUPPORT. FROM LEFT TO RIGHT, THE FIGURE CONTAINS A TEMPORAL SUPPORT COMPRISING THREE FRAMES, A TEMPORAL SUPPORT COMPRISING FIVE FRAMES, A "SIX-NEAREST-NEIGHBORS" SPATIO-TEMPORAL SUPPORT AND A CUBIC SPATIO-TEMPORAL SUPPORT.

$S = [p \in -1, 0, 1; q \in -1, 0, 1; l \in -1, 0, 1]$. For a temporal filter with $p = q = 0$, a typical support is $S = [p = q = 0; l \in -2, -1, 0, 1, 2]$. Some examples of temporal and spatio-temporal filter supports are shown in Figure 2.1. Note that the temporal extent determines the number of frame memories.

The differences between the various filters that can be described by (2.2) are attributed to the way in which the weights are established. In the simplest form, the filters have fixed weights with values that are globally optimal in a least-squares sense to estimate a constant value or a signal with known correlation immersed in noise. The resultant filters are then constrained versions of the original Wiener filter. In a more advanced form, the weights are adapted for each pixel on the basis of a segmentation procedure or least-squares minimization.

If all weights are given equal values within a temporal support, the filter results in a temporal averaging operator. A disadvantage of this filter is that it severely smoothes moving objects as reported by Huang and Hsu [6]. However, in non-moving areas such as the background, it is strongly noise suppressive. Temporal averaging was also investigated by Boyce [24].

Kalivas and Sawchuck [25] avoided smoothing of moving objects by the averaging filter by first using a segmentation step to classify objects and background. The support of the averaging filter was then restricted according to the shape of these objects. Effectively, this means a local adaptation consisting of zeroing the corresponding weights if the filter crosses a boundary. Temporal and spatio-temporal supports were considered. Although temporal blurring was avoided in both the temporal and spatio-temporal variants, the interior of the objects was over-smoothed by the spatio-temporal variants.

The contour plot method by Dekker *et al.* [26] is a method that assigns continuous values to the weights $w_{p,q,l}(i,j,k)$. It is based upon the same principle as the method of Kalivas and Sawchuck [25], namely to avoid filtering across object boundaries. The main feature of the method is the algorithm to check whether pixels belong to the same object or not. This is done by attaching an "altitude function" to the paths from the current pixel to all other pixels within the spatio-temporal support. The value of this altitude $a_{p,q,l}(i,j,k)$ measures the minimal effort of "climbing" from $g(i,j,k)$ to $g(i-p,j-q,k-l)$ along path π:

$$a_{p,q,l}(i,j,k) = \min_{\pi \in \Pi} \mathrm{U}\{g(i,j,k), g(i-p,j-q,k-l)\} \tag{2.3}$$

where Π is a limited set of "no-return" paths from (i,j,k) to $(i-p,j-q,k-l)$. The positive climbing effort $U\{\}$is found by considering the output of an edge detector. Note that if two pixel values are equal but do not belong to the same object, the effort $U\{\}$ is high and they are not regarded similar.

The conversion from altitude $a_{p,q,l}(i,j,k)$ to the filter weights $w_{p,q,l}(i,j,k)$ is done by means of the following function:

$$w_{p,q,l}(i,j,k) = \frac{c}{1 + \frac{a_{p,q,l}(i,j,k)^{\beta}}{\alpha}} \quad \beta \leq 0,\ \alpha < 0, \tag{2.4}$$

where c is for normalizing purposes and α and β are tuning constants. The results obtained by the contour plot method are good for sequences that are corrupted with a light to moderate level of noise. A too severe level of noise causes a degradation in the performance of the edge detector, which in turn will effect the quality of the filtered sequence [26].

The adaptive weighed-averaging filter by Özkan *et al.* [8, 27] also adapts the weights locally. The weights of the spatio-temporal filter are as follows:

$$w_{p,q,l}(i,j,k) = \frac{c}{1 + \alpha \mathrm{MAX}(\epsilon^2, (g(i,j,k) - g(i-p,j-q,k-l))^2)}, \tag{2.5}$$

where c is a normalizing constant, and α and ϵ are tuning parameters. It can be seen that if the absolute difference of the intensities between the current pixel $g(i,j,k)$ and another pixel $g(i-p,j-q,k-l)$ is less than ϵ, the pixel is included in the averaging. The value of the parameter ϵ depends on the noise variance.

The parameter α is a penalty parameter controlling how rapidly the weights should reduce as a function of the mismatch between pixel values. If $\alpha = 0$ the filter degrades to the temporal average; if α is large, then $1 + \alpha\epsilon^2 \approx \alpha\epsilon^2$ and the filter performs averaging only over the matching temporal values. Özkan *et al.* have investigated temporal [27] and spatio-temporal [8] supports in a motion-compensated context, and showed that the filter performs well for several values of α. A practical choice turned out to be $\alpha = 1$.

The weight vector $\boldsymbol{w}$, containing the weights, can also be assigned values that minimize the mean square estimation error. This is the class of constrained, i.e. limited support, Wiener filters. The values of the weights are first established by evaluating the statistical properties of the entire signal path. In a second pass these weights are used for filtering, thus making constrained Wiener filtering an off-line technique.

The following criterion function is used to find the globally optimal weights:

$$\boldsymbol{w} \leftarrow \min_{\boldsymbol{w}'} E\{\hat{f}(i,j,k) - f(i,j,k)\}^2. \tag{2.6}$$

Note that to solve (2.6), stochastic information about the original signal and noise is needed. A temporal filter of this form was investigated by Kleihorst *et al.* [28].

Equation (2.6) can be extended by incorporating quadratic terms which are able to more closely model non-stationary signals. The temporal version of this filter, known as a Volterra filter, looks like [29]:

$$\hat{f}(i,j,k) = \sum_{l \in S} w_l g(i,j,k-l) + \sum_{l,t \in S} b_{t,l} g(i,j,k-l) g(i,j,k-t). \tag{2.7}$$

The weight vectors $\boldsymbol{w}$ and $\boldsymbol{b}$ are also found by minimizing the mean square error as in Equation (2.6). To this end, up to 4^{th}-order statistics of the original signal and noise are necessary. Chan and Sullivan [30] proposed to use the Volterra filter for filtering clinical sequences in a spatio-temporal form following a motion-compensation step.

2.3 Order-statistic filters

Order-Statistic (OS) filters are variants of weighed-averaging filters. The distinction is that in OS filters the data is *ordered* before being used in the weighed averaging. Because of the ordering operation, correlation and time information are ignored in favor of magnitude information. The ordering action makes it possible to perform a very basic signal segmentation. Pixels belonging to the same object are usually automatically grouped. OS filters such as median filters are known for their edge-preserving properties [31] and are therefore often proposed for filtering the non-stationary image sequences. The linear OS filters have the following general structure:

$$\hat{f}(i,j,k) = \sum_{r=1}^{m} w_r(i,j,k) g_{(r)}(i,j,k), \tag{2.8}$$

where $g_{(r)}(i,j,k)$ are the ordered observations (ranks) from an observation window of odd size m centered at position (i,j,k). The filter weights are $\boldsymbol{w}(i,j,k)$, where $w_r(i,j,k)$ is the filter weight connected with *rank* r within the observation window.

The OS filters used in image-sequence filtering are mostly median filters. They range from the simple temporal median to multilevel median filters that can be designed to preserve certain image features. In later chapters we will consider other OS filters.

An early attempt to apply OS filters to the problem of image-sequence filtering was the straightforward temporal median filter proposed by Huang and Hsu [6] and by Naqvi *et al.* [32]. In this method, only the center weight $w_{\frac{m+1}{2}}(i,j,k)$ has a non-zero value, and the resulting filter is:

$$\hat{f}(i,j,k) = g_{(\frac{m+1}{2})}(i,j,k), \tag{2.9}$$

This filter has the advantage that it can be implemented in hardware if m is not too large [33, 34]. A disadvantage of the temporal median filter is that temporal "impulses", originating from moving thin structures are removed if their temporal support is less than half of the filter support [19, 35]. This causes artifacts which are called "edge busyness". Examples of such artifacts are shown in Figure 2.4 in the lower part of the right image, which was filtered with a temporal median filter. Parts of the moving characters and even entire characters are removed. The effect is even worse when seen in real-time video.

The artifacts of the temporal median filter can be avoided by the use of concatenated median filters, each with carefully designed spatio-temporal supports. Arce [36] and Alp and Neuvo [37] proposed methods which are based on multi-stage and multi-level spatio-temporal median filters, respectively.

A multi-stage median filter (MMF) is a method that combines the output of basic OS filters operating at the first stage of a cascaded filtering structure. The basic OS filters are designed to preserve specific features, such as lines, edges and object corners in a certain direction. By incorporating several sub-filters, basic image features in different orientations can be preserved. The type of feature to be preserved determines the subclass of the MMF. If the feature spans a 1D spatio-temporal line segment, a *unidirectional* support is employed. If the feature spans two line segments, each in orthogonal directions (for instance one in space, the other in time), a *bidirectional* support is employed.

Arce [36] considered two variants of the first stage. First a set of unidirectional median filters and second a set of bidirectional median filters (see Figure 2.2). The results of those sets of medians are used in the final stage. The final result is defined as the median value of the minimum and the maximum found in the first stage and the center pixel value.

$$\hat{f}(i,j,k) = \text{median}\{\max(\text{first stage}), g(i,j,k), \min(\text{first stage})\}. \tag{2.10}$$

Using bidirectional medians in the first stage resulted in a higher noise suppression.

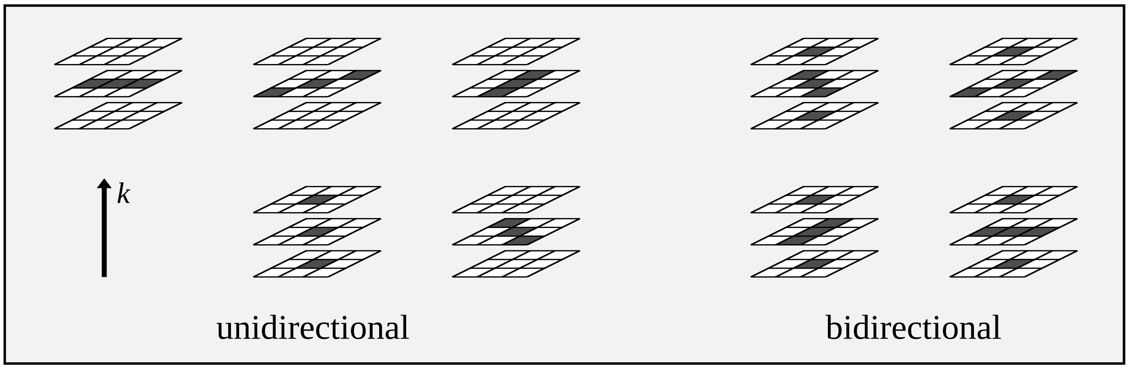


Figure 2.2: The MMF spatio-temporal sub-windows designed by Arce [36]

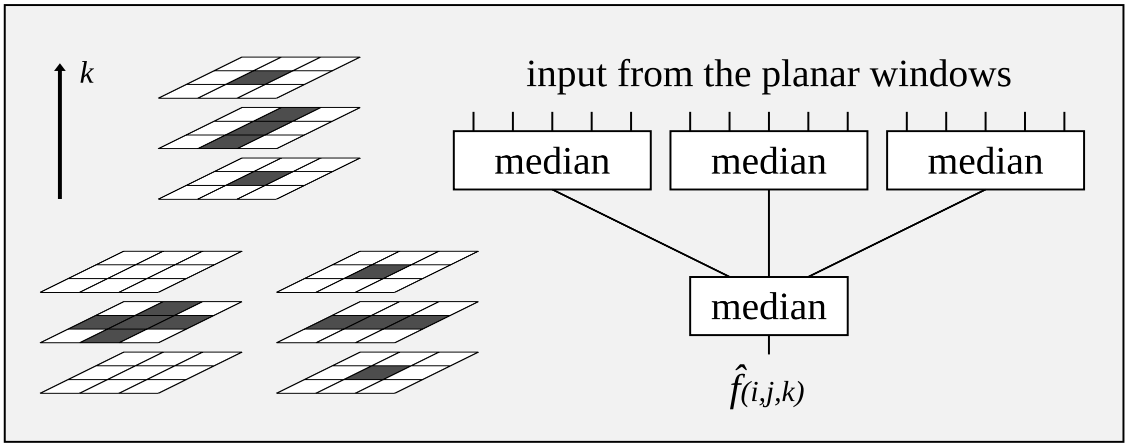


Figure 2.3: The planar multi-level median filter from Alp and Neuvo [37]; the orientation of bidirectional median filters on the left and the overall structure on the right

Kokaram and Rayner [38] experimented with a motion-compensated version of Arce's filter. Their aim was the removal of impulsive noise and they noted that a global operation of the filter would also distort image features. Therefore, they only applied the filter if impulse noise was detected in the current pixel. They concluded that their algorithm improved the result of Arce's, both in terms of quality of output and operation speed.

Alp and Neuvo [37] proposed the use of a multi-level median filter for the reduction of Gaussian noise. This filter will be described with the aid of Figure 2.3. Three 5-tap median filters operate on different planes in the 3D cube. A fourth median filter (3 taps) combines the outputs of the planar filters into a single result. This filter outperformed Arce's multistage filter in noise suppression [37].

Lee *et al.* [39] reported the use of the spatio-temporal median filters of Alp and Neuvo [37] in a motion-compensated environment. Their aim was to postprocess video sequences suffering from coding artifacts such as the "mosquito phenomenon"

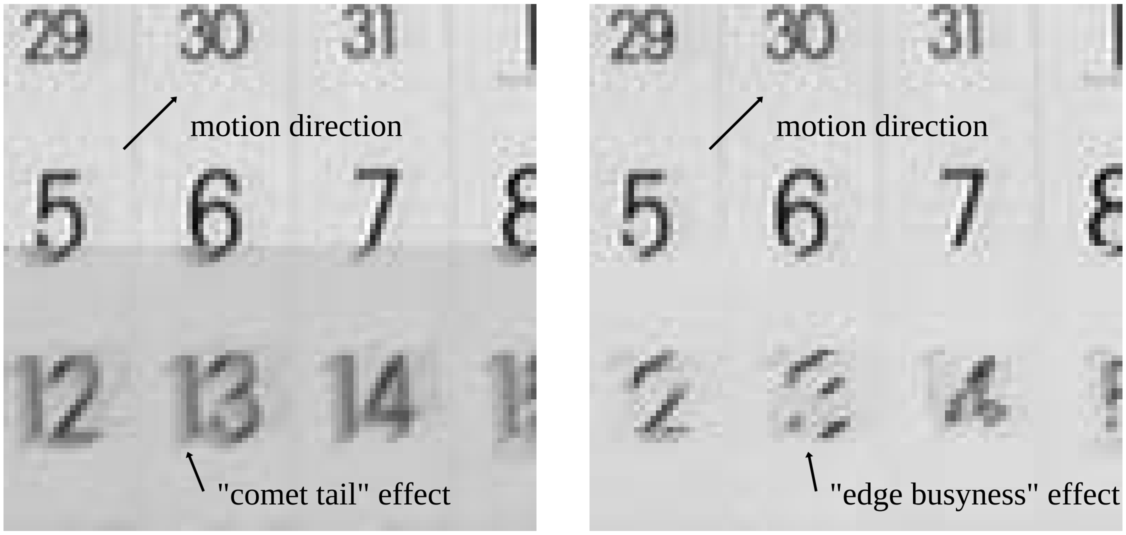


Figure 2.4: The lower half of the image on the left is processed with a recursive filter with a global control parameter. The "comet tail" effects are clearly visible with the moving characters. The lower half of the right image is processed with a temporal median filter. The filter removes certain moving line structures. In video, the effect is irregular, hence its name "edge busyness".

with inter-frame motion-compensated coding and the "blocking effect" with DCT-based schemes. The authors claim better results than Alp in dynamic scenes and comparable results in static scenes.

2.4 Switching filters

The Kalman filter is a popular choice for noise filtering time-series and images [40, 41]. For image sequences, a Kalman filter was considered by Cano and Bernard [42]. Their proposed 3D Kalman filter is given by:

$$\hat{\mathbf{S}}_a(i,j,k) = \hat{\mathbf{S}}_b(i,j,k) + \mathbf{K}(i,j,k)\left[\mathbf{G}(i,j,k) - \hat{\mathbf{S}}_b(i,j,k)\right], \qquad (2.11)$$

where $\hat{\mathbf{S}}_a(i,j,k)$ is the 3D global state vector of intensity values containing the current estimate after updating and $\hat{\mathbf{S}}_b(i,j,k)$ the state vector before updating. $\mathbf{G}(i,j,k)$ is a vector containing the current observation and $\mathbf{K}(i,j,k)$ is the 3D Kalman gain which is calculated for each pixel.

Cano and Benard avoided the processing burden of a full Kalman filter by separating the filter in a temporal and spatial part and regarding signal and noise as stationary signals. Their results indicate that the 3D Kalman filter can suppress the visibility of additive noise. However, the assumption of stationarity resulted in the introduction of unacceptable artifacts.

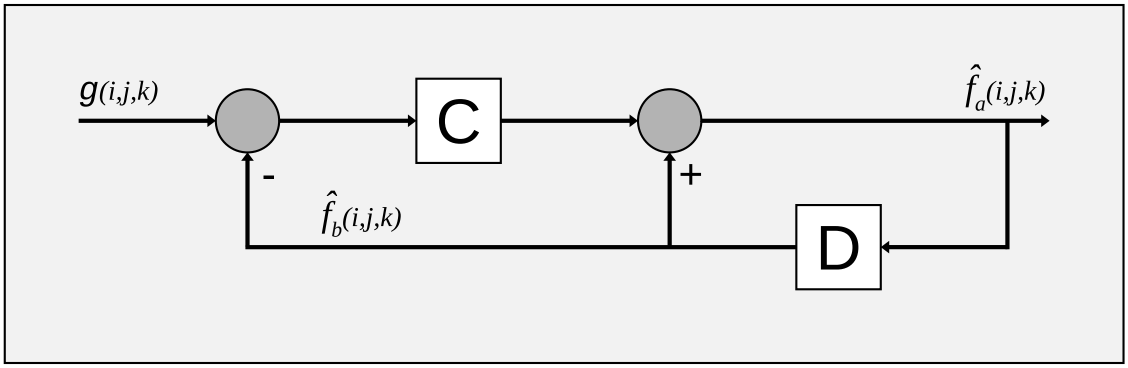


Figure 2.5: Among the nonlinear approaches without motion compensation this recursive signal-adaptive filter is often used. The current result is stored in the delay D, and is used in the next recursion pass. The actual differences in implementation are in the control parameter C.

Woods and Kim [43] avoided the computational burden associated with (2.11) by utilizing a "reduced update" Kalman filter which included motion compensation. Overall, despite actions to relieve the computational effort, it appears that the calculation of the Kalman gain and the state transitions for each pixel is considerable.

The following simplified Kalman-like structure is computationally more efficient, and was therefore substantially exploited [7, 44, 45, 46, 47, 48]:

$$\hat{f}_a(i,j,k) = \hat{f}_b(i,j,k) + C(i,j,k)[g(i,j,k) - \hat{f}_b(i,j,k)]. \tag{2.12}$$

Here $\hat{f}_b(i,j,k)$ is the estimate "before updating" and $\hat{f}_a(i,j,k)$ the final estimate "after updating". The estimate before updating is often chosen as [7, 44, 45, 47]:

$$\hat{f}_b(i,j,k) = \hat{f}_a(i,j,k-1), \tag{2.13}$$

which implies a recursion in the temporal direction. This is a great advantage compared to the weighed-averaging filters, as it requires only 1 frame memory. The relatively simple structure of the filter is illustrated in Figure 2.5. By adapting the value of control parameter $C(i,j,k)$, it is possible to instantaneously "switch off" or "switch on" the filter. For instance, if $C(i,j,k) = 1$ the observation $g(i,j,k)$ is forwarded; however, if $C(i,j,k) = 0$ then the corresponding estimate in the previous frame is forwarded. Any value between these extremes can be used. An advantage of the switching filter is that the response of the filter can be changed by varying a single parameter. A disadvantage of this type of adaptation is that the noise is not reduced when $C(i,j,k) = 1$.

The actual differences between the several proposals based on Equation (2.12) lie in the choice of the control parameter $C(i,j,k)$. They range from fixed, discrete values to continuous values which are for instance found by locally minimizing the mean square estimation error.

The simplest case is to fix $C(i,j,k)$ globally to a constant value. Dennis [49] reports of an experiment with $C = 0.25$. The long step-response of this filter, however, causes distortion of moving objects in the sequence. The resulting visual effect is called the "comet-tail" effect and is visible in Figure 2.4.

A large difference between the current observation and the previous filter output is usually due to motion or scene changes. This difference can be used for locally adapting $C(i,j,k)$. In this case, the value of $C(i,j,k)$ depends on the "prediction error" $e(i,j,k) = g(i,j,k) - \hat{f}_b(i,j,k)$. This causes the filter response to adapt instantaneously to the characteristics of the signal. In this light, McMann [44] used the following approach:

$$C(i,j,k) = \begin{cases} 0 & \text{if} \quad |e(i,j,k)| \leq \tau, \\ 1 & \text{if} \quad |e(i,j,k)| > \tau, \end{cases} \tag{2.14}$$

where τ is an appropriately chosen threshold which depends on the noise deviance.

A finer adaptation is used by Dubois and Sabri [7], Reinen [50] and Dubois [47], who used the following double threshold function for the control parameter:

$$C(i,j,k) = \begin{cases} \alpha & \text{if} \quad |e(i,j,k)| \leq 3; \\ \frac{1-\alpha}{7}|e(i,j,k)| - 10\alpha + 3 & \text{if} \quad 3 < |e(i,j,k)| \leq 10; \\ 1 & \text{if} \quad |e(i,j,k)| > 10. \end{cases} \tag{2.15}$$

Practical values for α are $0.1 \cdots 0.2$. The filter was used in combination with motion compensation [7, 47]. Reinen [50], also used a 2^{nd} order variant of this filter for application to clinical image sequences.

A continuous adaptation of the control parameter was suggested by Crawford [45]:

$$C(i,j,k) = 1 - exp\left\{\frac{|e(i,j,k)|}{\tau}\right\}^{\gamma}, \tag{2.16}$$

where τ and γ are tuning constants. Again, 2^{nd} order variants of the filter were proposed.

The recursive filter from (2.12) can be improved by using a priori information of the image sequence model, for instance in a Kak filter. Motivated by its earlier success in image and speech processing, the Kak filter was investigated in a 3D variant by Triplicane [51] and Katsaggelos *et al.* [52]. The 3D Kak filter assumes a 3D AR model for the original signal:

$$f(i,j,k) = \sum_{p,q,l \in A} a(p,q,l) f(i-p, j-q, k-l) + v(i,j,k). \tag{2.17}$$

Here, $v(i,j,k)$ is a white, signal-independent driving noise and $a(p,q,r)$ are the fixed model coefficients. The order of the model is determined by its support A.

The effect of the image model on the filter (2.12) manifests itself in the estimate before updating:

$$\hat{f}_b(i,j,k) = \sum_{p,q,r \in S} a(p,q,r)\hat{f}_a(i-p, j-q, k-r). \tag{2.18}$$

Note the similarity with the update equation (2.13) discussed earlier if $a(0,0,1) = 1$ and all other model coefficients are zero. Triplicane [51] and Katsaggelos *et al.* [52] used a fixed control parameter $C(i,j,k)$, the value of which was optimal in mean square estimation-error sense, given the image models and the noise statistics.

Katsaggelos *et al.* [53] also considered concatenated unidirectional Kak filters. In this method, the final estimate is given by:

$$\hat{f}(i,j,k) = \rho_t \hat{f}(i,j,k-1) + C(k)(\hat{f}_j(i,j,k) - \rho_k \hat{f}(i,j,k-1)), \tag{2.19}$$

where:

$$\hat{f}_j(i,j,k) = \rho_v \hat{f}(i,j-1,k) + C(j)(\hat{f}_i(i,j,k) - \rho_j \hat{f}(i,j-1,k)), \tag{2.20}$$

and:

$$\hat{f}_i(i,j,k) = \rho_h \hat{f}(i-1,j,k) + C(i)(g(i,j,k) - \rho_i \hat{f}(i-1,j,k)). \tag{2.21}$$

Here, $C(i), C(j)$ and $C(k)$ are the horizontal, vertical and temporal control parameters, respectively, and ρ_h, ρ_v and ρ_t are the horizontal, vertical and temporal correlation coefficients.

Non-adaptive and an adaptive versions of this filter have been proposed in [53]. In the non-adaptive version, the correlation coefficients and global control parameters were globally fixed to optimal values. As this tends to blur spatio-temporal edges, an adaptive version was designed. In this version, the correlation coefficients and control parameters are controlled by spatio-temporal edge detectors.

A relatively popular filter structure arises when the "before update" estimate is replaced by an estimate of the local mean of the observation [46, 48]:

$$\hat{f}_b(i,j,k) = \hat{\mu}_g(i,j,k). \tag{2.22}$$

Note that by this choice, the filter structure is not longer recursive. It will be shown in Chapter 5 that the optimal Minimum Mean Square Error (MMSE) control parameter for (2.22) is:

$$C(i,j,k) = 1 - \hat{\sigma}_n^2 / \hat{\sigma}_g^2(i,j,k). \tag{2.23}$$

With the use of this control parameter, estimates of the noise variance σ_n^2 and the local variance of $g(i,j,k)$, denoted by $\sigma_g^2(i,j,k)$, have to be established. The combination of (2.12), (2.22) and (2.23) is called the "Local Linear MMSE" (LLMMSE) filter. It is used in combination with motion compensation by Sezan *et al.* [46]. They used temporal "box averages" for $\hat{\sigma}_g^2(i,j,k)$ and $\hat{\mu}_g(i,j,k)$, which are inaccurate if the "box" crosses object boundaries. The effect is that the estimate of local variance is too high which causes the control parameter to switch off filtering, yielding sharp but noisy spatio-temporal edges.

Martinez and Lim [54] tried to avoid this undesirable effect in the LLMMSE filter by performing an implicit form of motion estimation and compensation. They estimated $\hat{\sigma}_g^2(i,j,k)$ and $\hat{\mu}_g(i,j,k)$ by "box averages" from unidirectional windows which were directed to mimic certain motion trajectories of the current pixel. They distinguished between no-motion and four types of possible motion: left, right, up, and down translation.

The filtering was performed for each direction in a concatenated way. In case of operating in a stationary part of the sequence, all 5 filters contribute to the final result, giving a maximum spatio-temporal span. If the concatenated filter crosses moving objects, the estimator which is oriented along the motion trajectory will provide the final result, as its window has the correct estimate for $\sigma_g^2(i,j,k)$ and all other filters are switched off by their respective control parameters.

Kleihorst *et al.* [48, 55] used robust estimators based on order statistics to avoid the inaccuracy at object boundaries of "box averages". Both non motion-compensated spatio-temporal estimators [48] and motion-compensated recursive spatio-temporal estimators [55] were considered. These LLMMSE filters will be the subject of discussion in Chapter 5.

2.5 Signal decomposition

A classical way to handle the filtering of non-stationary signals is to decompose the signal into a non-stationary and a homogeneous part by trend-removal and normalization. If the noise is a stationary signal it is entirely mapped into the homogeneous part. Any linear noise filter can then be used for the filtering this part.

Signal decomposition involves the estimation of local signal statistics. Kleihorst *et al.* [28] and Katsaggelos *et al.* [53] have used adaptive order-statistic based estimators to estimate the local mean $\mu_g(i,j,k)$ and local deviation $\sigma_g(i,j,k)$ from the observed signal. With the estimated values of these statistics, the non-stationary signal can be decomposed into a non-stationary part consisting of $\hat{\mu}_g(i,j,k)$ and $\hat{\sigma}_g(i,j,k)$ and a homogeneous part $y(i,j,k)$:

$$y(i,j,k) = \frac{g(i,j,k) - \hat{\mu}_g(i,j,k)}{\hat{\sigma}_g(i,j,k)}, \tag{2.24}$$

If we denote the actual noise filtering operation by $F\{\}$, the result of the decomposition method can be written as follows:

$$\hat{f}(i,j,k) = \hat{\mu}_g(i,j,k) + \hat{\sigma}_g(i,j,k)\; F\left\{\frac{g(i,j,k) - \hat{\mu}_g(i,j,k)}{\hat{\sigma}_g(i,j,k)}\right\}. \tag{2.25}$$

Note the similarity of the overall structure with the switching filters from the previous section.

Kleihorst *et al.* [14] and Katsaggelos *et al.* [53] used a simple, causal weighed-averaging filter for noise filtering of $y(i,j,k)$:

$$F\{y(i,j,k)\} = \sum_{l \in S} w_{0,0,l}(i,j,k)\, y(i,j,k-l). \tag{2.26}$$

The components of the weight vector $\boldsymbol{w}(i,j,k)$ are found by recursively minimizing the cumulative square error:

$$\boldsymbol{w}(i,j,k) \;\leftarrow\; \min_{\boldsymbol{w}'(i,j,k)} \left\{ \sum_{p=1}^{k} \lambda^{k-i} \left(y(i,j,p) - F\{y(i,j,p)\}\right)^2 \right\}, \tag{2.27}$$

where the updating is performed along the temporal direction, and λ serves as a forgetting factor [56]. The reason for using the cumulative updating including the forgetting factor, is to adapt to the auto-correlation of $y(i,j,k)$. Although this signal is homogeneous, it does not necessarily have a fixed auto-correlation.

This approach was investigated using temporal [28, 53] and spatio-temporal estimators [19] for estimating the local statistics. The decomposition method, and in particular the estimation method, are investigated more thoroughly in Chapter 4.

2.6 Bayesian approaches

A number of methods were proposed for filtering image sequences that are based upon the maximization of a likelihood function. More formally:

$$\hat{f}(i,j,k) \;\leftarrow\; \max_{f'(i,j,k)} p\{g(i,j,k) | f'(i,j,k)\}. \tag{2.28}$$

The estimator obtained in this way finds the most probable original image that has caused the observed signal. Maximum likelihood or Bayesian approaches are often applied in image restoration and identification [12, 15] and motion estimation [18]. Likelihood methods strongly depend on stochastic observation and noise models. This is in contrast with many of the previous methods, where it was tacitly assumed that the noise was Gaussian.

By using edge-dependent weighing in the criterion function, Bayesian approaches are able to handle non-stationary signals. Because of the complexity of the likelihood function, maximization of (2.28) is usually performed by iterative procedures.

Hong and Brzakovic [57] have investigated a probabilistic approach by modeling image sequences as Markov Random Fields (MRF) [12]. The specific problem they were trying to solve was:

$$\hat{\boldsymbol{f}}(k) \leftarrow \max_{\boldsymbol{f}'} p\{\boldsymbol{f}'|\boldsymbol{g}(k), \hat{\boldsymbol{f}}(k-1)\}, \tag{2.29}$$

where $\boldsymbol{g}(k)$ and $\hat{\boldsymbol{f}}(k)$ denote the observed and estimated frame k. Using Bayes' theory and the Markov Random Field (MRF) property (the current pixel only depends on neighboring pixels), (2.29) can be written for pixel values instead of full frames as:

$$\hat{f}(i,j,k) \leftarrow \max_{f'(i,j,k)} p\{g(i,j,k)|f'(i,j,k)\}\ p\{f'(i,j,k)|\text{ neighboring pixels}\}. \tag{2.30}$$

Assuming that the additive noise is white and Gaussian, the first part of (2.30) can be written as:

$$p\{g(i,j,k)|f'(i,j,k)\} = \frac{1}{\sigma_n\sqrt{2\pi}} \exp\left\{\frac{-(g(i,j,k)-f'(i,j,k))^2}{2\sigma_n^2}\right\}. \tag{2.31}$$

Using the properties of MRFs, the second part in (2.30) can be written as a "Gibbs" distribution [58]:

$$p\{f'(i,j,k)|\text{ neighboring pixels}\} = \frac{1}{\kappa}\exp\left\{-U\{f'(i,j,k), \hat{f}(i,j,k-1)\}\right\}. \tag{2.32}$$

Here, κ is a normalizing constant and $U\{\}$ is an "energy function".

The actual estimate is found by solving Equation (2.30). Hong and Brzakovic [57] solved this likelihood function iteratively with the "Iterated Conditional Modes" (ICM) approach.

Geman and McClure [59] discuss an approach based on regularized minimization of the following, more general criterion function:

$$\hat{f}(i,j,k) \leftarrow \min_{f'(i,j,k)} \alpha U\{f'(i,j,k)\} + V\{f'(i,j,k), g(i,j,k), g(i,j,k-1)\}. \tag{2.33}$$

It involves two terms of which the relative levels of importance are weighed by α. The functional $U\{f'(i,j,k)\}$ incorporates the image model and assigns high values to undesirable estimated images. In effect, it compromises a function that penalizes large 2^{nd} order derivatives.

The functional $V\{f'(i,j,k), g(i,j,k), g(i,j,k-1)\}$ incorporates the observation model. It denotes the relation between the candidate estimate and the observed

Table 2.1: Summary of the filters implemented from this chapter

Filter	Equation	Reference	Description
average	(2.2)	[6]	Temporal averaging filter
Naqvi	(2.9)	[32]	Temporal median filter
Arce	(2.10)	[36]	Spatio-temporal median filter
3D Kak	(2.18)	[51, 52]	Least-squares adaptive filter
Kleihorst	(2.12),(2.22),(2.23)	[48]	Locally-adaptive filter
Martinez	(2.12),(2.22),(2.23)	[54]	Motion-tracking filter

data. It penalizes candidates $f'(i, j, k)$ that have not likely resulted in the observation.

The functional in (2.33) was solved iteratively for each pixel. Geman and McClure [59] have used this algorithm in a motion-compensated form to restore motion picture material. Their aim was to correct for scratches and accumulation of dirt on film material.

2.7 Experimental evaluation

We compared the performances of several filters described in this chapter. The selected filters are listed in Table 2.1. Included are representative methods from most classes with the exception of the Bayesian method, which will be evaluated in Chapter 3 and the decomposition method which is the subject of Chapter 4. The methods that have been evaluated in this chapter were all reported in literature to operate without motion compensation.

From the weighed-averaging class we have selected the temporal averaging of Huang and Hsu [6]. The class of OS filters is represented by the temporal median filter of Naqvi *et al.* [32] and the spatio-temporal multi-stage median filter by Arce [36]. From the switching filters we have tested the recursive 3D Kak filter by Triplicane [51] and Katsaggelos *et al.* [52], and the non-recursive OS-supported LLMMSE filter by Kleihorst *et al.* [48]. In addition, we have investigated the concatenated motion-tracking LLMMSE filter by Martinez and Lim [54] that contains implicit motion compensation, as discussed in Section 2.4. The “3D Kak” filter was evaluated at the Digital Signal and Image Processing Laboratory of Northwestern University, Illinois.

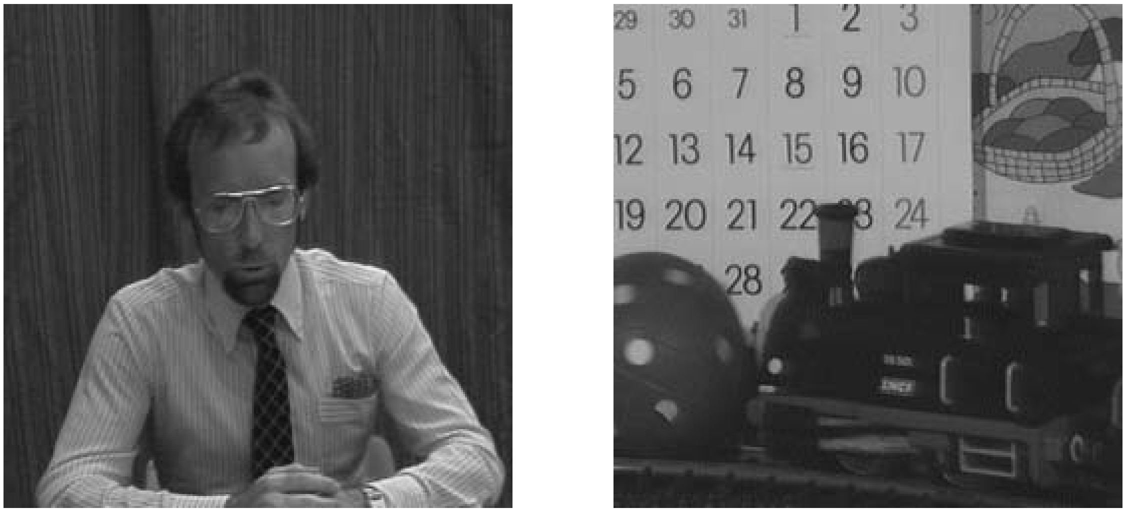


FIGURE 2.6: FRAME 21 OF THE "TREVOR" SEQUENCE ON THE LEFT AND FRAME 1 OF THE "MOBILE" SEQUENCE ON THE RIGHT.

Each filtering technique listed in Table 2.1 was applied to noisy versions of the image sequences "mobile" (frames 1-40) and "Trevor" (frames 21-70). These image sequences are generally used for evaluating filtering and coding methods. We have selected representative sections with image sizes of 256 × 256 pixels from the original sequences. The "Trevor" sequence contains a lot of jerky motion whereas the "mobile" sequence contains (predictable) smooth motion. The image data in "mobile" has a more detailed nature than the "Trevor" sequence. Frames of both sequences are shown in Figure 2.6.

The sequences were degraded using simulated additive, white Gaussian noise up to signal-to-noise ratios (SNRs) of 20 and 10 decibels (dB). The expression used for defining this SNR, already stated in Chapter 1, is repeated here:

$$\mathrm{SNR} = 10\log_{10}\left\{\frac{\sigma_f^2}{\sigma_n^2}\right\}\ (\mathrm{dB}). \tag{2.34}$$

Here, σ_f^2 is the variance of the original signal and σ_n^2 is the variance of the additive noise. To evaluate the performance of each filtering technique we use the *improvement in SNR* per frame, which is defined as

$$\mathrm{SNRi}(k) = 10\log_{10}\left\{\frac{\sum_{i=1}^{I}\sum_{j=1}^{J}[f(i,j,k)-g(i,j,k)]^2}{\sum_{i=1}^{I}\sum_{j=1}^{J}[\hat{f}(i,j,k)-f(i,j,k)]^2}\right\}\ (\mathrm{dB}), \tag{2.35}$$

where I, J are the spatial dimensions. Although it has been argued that this type of metric is a poor measure of the true visual image quality, we have decided to use it because of the lack of any other widely accepted metric. This does not mean that limited effort is devoted to other metrics as can be appreciated from [60, 61, 62, 63]. In addition to the SNRi measure, we will also give some visual impressions of the filter performance.

Table 2.2: Average *improvement in* SNR for the “mobile” sequence.

Filter	*Average Improvement in* SNR (range)	
	SNR=10dB	SNR=20dB
average	2.8 (0.6)	-3.5 (1.4)
Naqvi	2.0 (0.5)	-2.7 (1.7)
Arce	3.5 (0.2)	1.1 (0.5)
3D Kak	4.7 (0.5)	1.0 (0.0)
Kleihorst	7.4 (0.2)	3.9 (0.1)
Martinez	4.1 (0.3)	2.2 (0.3)

Table 2.3: Average *improvement in* SNR for the “Trevor” sequence.

Filter	*Average Improvement in* SNR (range)	
	SNR=10dB	SNR=20dB
average	3.4 (1.2)	-1.8 (3.1)
Naqvi	2.2 (0.9)	-2.8 (2.6)
Arce	3.9 (0.2)	2.1 (0.6)
3D Kak	5.0 (1.3)	1.7 (0.3)
Kleihorst	7.2 (0.4)	2.8 (0.3)
Martinez	4.3 (1.1)	0.4 (0.9)

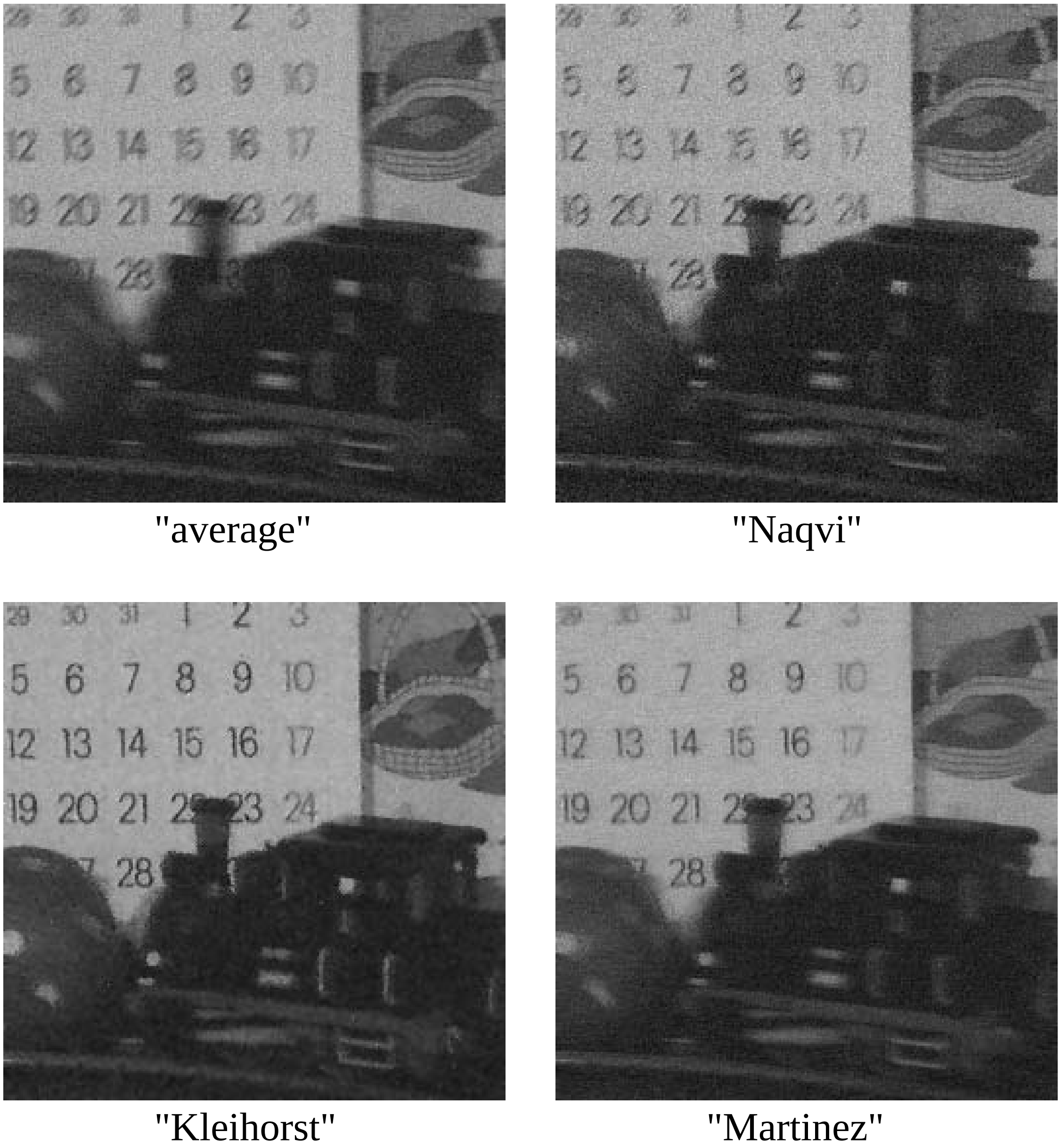
The experimental results are summarized in Figures 2.8 and 2.9 and Tables 2.2 and 2.3. In the tables, the average improvement per frame and the range (between parentheses) of the improvements are presented. We have avoided the influence of “start-up” or “shut-down” effects of certain filters in the average improvement metric and the range. From the results we can deduce that methods that take non-stationarities explicitly into account, such as the filter “Martinez” and “Kleihorst” outperform methods which assume stationary signals such as “average”.

The spatio-temporal filters such as “3D Kak”, “Arce”, “Kleihorst” and “Martinez” that can exploit more data than the temporal filters have an overall larger improvement.

The response to motion can be most clearly seen from the improvement in the “Trevor” sequence. An irregular curve indicates either deterioration of moving objects or a (legitimate) filter shutdown of the switching filters. The irregular curve generated by the non switching “average” and “Naqvi” filters are caused by deterioration of moving objects. The irregularities caused by “Martinez” are the result of a diminishing filter support in non-stationary regions.

Filters that assume stationary signals have more difficulties in filtering non-stationary signals at high SNRs. Ideally, a filter should not affect the original signal, but it is clear from the tables that filters such as “average” and “Naqvi” are distorting the original signal at 20dB SNR.

From a visual point of view, the “average” filter tremendously deteriorates the signal. The “Naqvi” and “Arce” filters are not very noise-suppressive and the “3D Kak”,“Martinez” and “Kleihorst” filter yield an appreciable image quality improvement. Some results from “average”,“Naqvi”,“Kleihorst” and “Martinez” are shown in Figure 2.7. The filter results in this chapter can be used as reference material with respect to the methods dealt with in the remaining chapters, where we will consider motion compensation and adaptive order-statistic filters.



"average"

"Naqvi"

"Kleihorst"

"Martinez"

Figure 2.7: Filter results for 10dB SNR of some specific filters

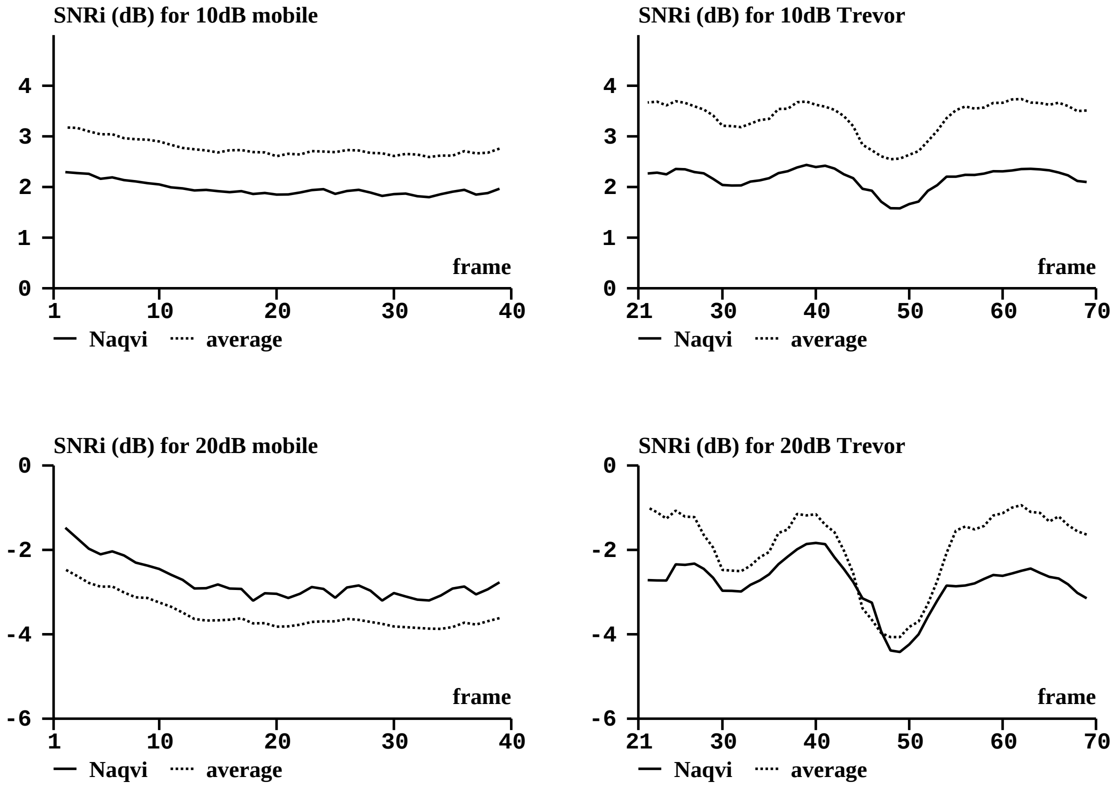


FIGURE 2.8: EXPERIMENTAL RESULTS FOR THE NON-MOTION-COMPENSATED TEMPORAL FILTERS.

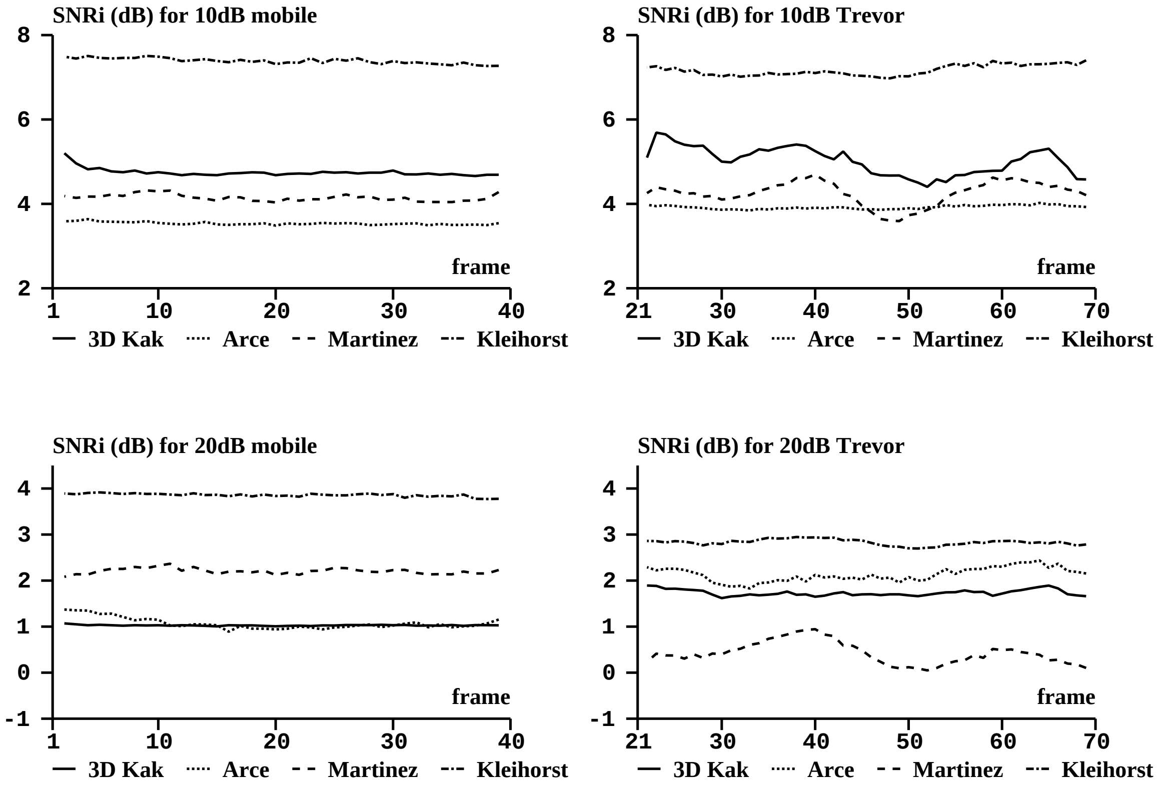


Figure 2.9: Experimental results for the non-motion-compensated spatio-temporal filters.

Chapter 3

Motion-compensated filtering

3.1 Introduction

In the Chapter 2 we have seen that many filtering methods rely on adaptivity to handle (temporally) non-stationary signals. As the temporal non-stationarities are often caused by object motion, they can be avoided in these cases by filtering along the motion trajectories. For instance, instead of the familiar expression for a (recursive) filter $F\{\}$:

$$\hat{f}(i,j,k) = F\{\hat{f}(i,j,k-1)\}, \tag{3.1}$$

we use a motion-compensated form:

$$\hat{f}(i,j,k) = F\{\hat{f}(i-d_i, j-d_j, k-1)\}. \tag{3.2}$$

where d_i and d_j represent the vertical and horizontal displacement.

The filtering algorithm can now be described as a three-step process as illustrated in Figure 3.1. First, in a motion-estimation part, the object motion is estimated from the observed signal. Second, the temporal signal is compensated for motion based on the motion estimates $\hat{\boldsymbol{d}}(i,j,k)$. The third part comprises the noise filter.

Compensation for motion improves the filtering results of any noise filter which is (partly) active in the temporal direction. Filters that hardly manage non-stationary signals will definitely benefit because signal distortion is avoided. Filters that manage non-stationary signals will benefit, although less, because the motion compensation results in an effectively larger homogeneous section of the filter support.

In general, object motion is estimated from the noisy observation. Depending on the robustness of the motion estimator, the noise may result in inaccurate motion

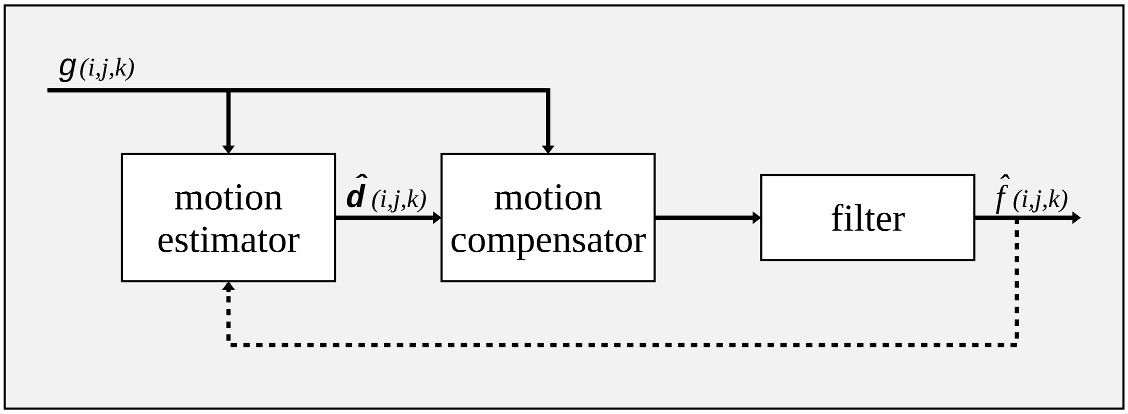


Figure 3.1: A motion-compensated filtering scheme consists of three parts: First, the motion is estimated, second, the observed temporal signal is compensated for motion, and third, it is filtered.

estimates [6, 13, 64]. This causes colored noise in the compensated temporal signal which will aggravate the final noise filtering process. The effects are annoying patches in the filter result. The danger of temporal coloring depends on the SNR and is more severe in the low-variance areas such as the background where the *local* SNR is low.

Early motion-compensated filtering techniques relied on motion estimators that were originally designed for motion-compensated coding purposes [6, 7]. Among those are the gradient and pel-recursive techniques and the full-search block-matching algorithm [2]. Those estimation methods could not handle the noise very well. The application area of motion-compensated noise filtering was therefore initially limited to relatively high SNRs.

Later on, more effort was devoted to decreasing the noise sensitivity of motion estimators. Suggestions arising from this effort are for instance pre-filtering of the noisy sequence [52] and modifying the structure of the motion-estimation algorithm [55].

Motion estimation and estimation of the original sequence can be regarded as coupled problems. This point of view has resulted in simultaneous estimation approaches, where the object motion and the original sequence are estimated jointly by a single estimator [18].

In this chapter we will discuss motion estimation and compensation as a means of *circumventing* non-stationarities in temporal signals. First, in Section 3.2 we consider various ways to compensate the temporal signal for motion, based upon the filter support and the motion estimates available. In Section 3.3 we deal with the use of well-known motion-estimation techniques on noisy data. In Section 3.4 we discuss reducing the noise sensitivity of motion-estimation methods. Here, we also consider the simultaneous approaches that regard estimation of original signal and noise as coupled problems. Finally, Section 3.5 gives an experimental evaluation of a noise-robust motion-estimation algorithm and some representative motion-compensated filtering methods.

3.2 Motion-compensation strategies

Motion compensation in image sequence filtering can be performed in a number of ways depending on demands related to the filter support and the assumptions on the motion. This section gives an overview.

The motion-compensation stage in Figure 3.1 uses the object motion vector calculated by the motion estimator to compensate the sequence. Often, the temporal support of a noise filter covers multiple frames in which case additional motion vectors are necessary for compensation. These additional motion vectors can be established in two ways. First, additional motion estimates can be used or second, previous motion estimates are simply copied to avoid the computational burden involved with motion estimation. The decision to copy the previous motion estimate is based upon the assumption that the object moves in a non-accelerated, translational fashion. If the assumption fails in a specific situation, compensation errors result.

The simplest compensation occurs in 1^{st} order temporally recursive filters that only depend on filter results of the previous frame and the current pixel. We have already given an example in Equations (3.2) and (3.1). Here, only the motion estimated from frame k to frame $k-1$ is used in compensation (Figure 3.2a).

Non-causal filters such as weighed average and LLMMSE filters also exploit data from future frames such as $k+1$. Compensation can be performed in two ways. One way is to assume non-accelerated motion, in which case no additional motion estimates are performed but symmetrically extended vectors are used. Then, the motion vectors describing displacements from frame $k-1$ to k are simply applied to frame $k+1$ (Figure 3.2b). The other way is not to assume non-accelerated motion. Then, an additional motion estimate from frame k to frame $k+1$ is used for compensation (Figure 3.2c). The latter method has a more accurate compensation.

Filter structures using an even larger temporal support such as some temporal 2^{nd} order recursive filters [45, 50], non-recursive temporal switching filters [46, 55] and temporal weighed-averaging filters [6, 8] need additional compensation, for instance to frame $k+2$. The displacements necessary for this additional compensation can be derived in several ways. The least expensive way is by assuming non-accelerated motion and extending previous motion estimates (Figure 3.2d). In a more complex

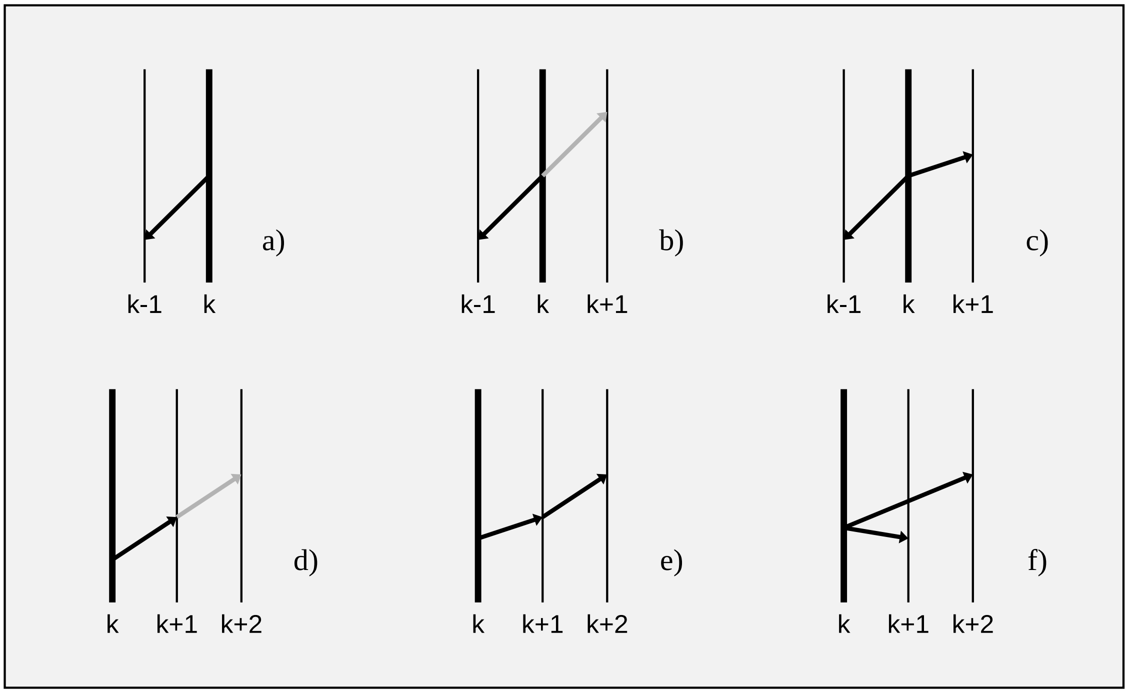


Figure 3.2: Motion compensation can be implemented in a number of ways depending on complexity considerations and assumptions. 1^{st} order recursive filters need only one motion estimate (dark vector) to the previous frame as in a). In other simple schemes no motion estimate is performed but the available motion estimate is applied to adjacent frames, b) or extended d) (light vectors). In more advanced methods the motion is estimated from the current frame to all frames used by the filter, c), e) and f).

situation, additional motion estimates are retrieved between frames k and $k+1$ and frames $k+1$ and $k+2$ as in Figure 3.2e or from the current frame to each future frame exploited by the filter as in Figure 3.2f. This is a complex method. The motion estimates relative to the previous frames can be established in a similar manner. Usually, stored results from previous frames can be employed.

3.3 Motion estimation from noisy sequences

Several motion-estimation algorithms are used in image-sequence analysis, motion-compensated coding, sub-sampling and filtering. The methods can be classified into several categories [65, 66]: segmentation-based methods; transform-domain methods such as phase-plane correlation; matching methods including block-matching; gradient methods such as pel-recursive motion estimation and statistical methods including maximum likelihood and Bayesian approaches.

Early motion-compensated filtering techniques relied on motion-estimation techniques derived for coding purposes, namely matching and gradient techniques. After deriving the motion model used in these methods, we consider the influence of noise on their behavior.

3.3.1 The motion model

All motion estimators relate to a model for the object motion. Most motion models are related to the *optical flow model* which assumes that brightness changes are only due to motion. The resulting optical flow equation is based on the assumption that the object brightness is constant over time along the motion trajectory [67, 68]:

$$\frac{df(i,j,k)}{dk} = 0, \tag{3.3}$$

where i, j and k are now considered as continuous variables. The intensity $f(i,j,k)$ is a function of the coordinates i, j and time k. Therefore, rewriting (3.3) gives:

$$\frac{\delta f(i,j,k)}{\delta i}\frac{d_i}{dk} + \frac{\delta f(i,j,k)}{\delta j}\frac{d_j}{dk} + \frac{\delta f(i,j,k)}{\delta k} = 0, \tag{3.4}$$

where d_i/dk and d_j/dk are the components of the optical flow which are regarded as the motion parameters. They can be estimated if the spatial gradients $\delta f(i,j,k)/\delta i$ and $\delta f(i,j,k)/\delta j$ are known. Setting $\delta k = 1$ and $\delta f(i,j,k)/\delta k = f(i,j,k) - f(i,j,k-1)$ will then yield the motion estimates. Effectively, the following motion model is assumed:

$$f(i,j,k) = f(i-d_i, j-d_j, k-1). \tag{3.5}$$

This relation shows that the motion between subsequent frames is locally translational, i.e. can be entirely described by the motion vector $[d_i, d_j]^T$. As a result, occlusion, variations in object intensity, and non-translational motion such as (3D) rotation, are not covered.

Commonly, in motion-compensated noise-filtering, the motion vectors are estimated from the noisy sequence. This implies that the motion model of Equation (3.5) is used on the observation $\boldsymbol{g}$ resulting in:

$$g(i,j,k) = g(i-d_i, j-d_j, k-1) + n(i,j,k) - n(i-d_i, j-d_j, k-1). \tag{3.6}$$

The noise terms corrupt the motion model and may cause inaccurate motion estimates.

3.3.2 Gradient techniques

Gradient techniques for motion estimation can be derived by specifying the frame difference $\Delta g(i,j,k)$ as follows:

$$\Delta g(i,j,k) = g(i,j,k-1) - g(i,j,k). \tag{3.7}$$

Substitution of Equation (3.6) results in:

$$\begin{aligned} \Delta g(i,j,k) \quad &= \quad g(i,j,k-1) - g(i-d_i, j-d_j, k-1) \\ &\quad\quad +n(i-d_i, j-d_j, k-1) - n(i,j,k). \end{aligned} \tag{3.8}$$

This relation can be linearized in d_i and d_j as follows:

$$\Delta g(i,j,k) = \frac{\delta g(i,j,k-1)}{\delta i} d_i + \frac{\delta g(i,j,k-1)}{\delta j} d_j + \epsilon(i,j,k), \tag{3.9}$$

here, $\frac{\delta g(i,j,k-1)}{\delta i}$ and $\frac{\delta g(i,j,k-1)}{\delta j}$ are the spatial gradients and $\epsilon(i,j,k)$ represents the approximation error.

Assuming that the motion vector, which will be denoted as $\boldsymbol{d} = [d_i, d_j]^T$, is equal for a number of neighboring pixels, (3.9) can be expanded and written in a matrix-vector form:

$$\boldsymbol{\Delta g} = \boldsymbol{G}\boldsymbol{d} + \boldsymbol{\epsilon}. \tag{3.10}$$

Here, $\boldsymbol{G}$ is a matrix containing the spatial gradients estimated from the observed sequence. The motion vector is found by solving:

$$\hat{\boldsymbol{d}} \leftarrow \min_{\boldsymbol{d}'} ||\boldsymbol{\Delta g} - \boldsymbol{G}\boldsymbol{d}'||^2, \tag{3.11}$$

Usually, the problem is over-determined and the motion vector is found using the generalized inverse of $\boldsymbol{G}$:

$$\hat{\boldsymbol{d}} = [\boldsymbol{G}^T\boldsymbol{G}]^{-1}\boldsymbol{G}^T\boldsymbol{\Delta g} \tag{3.12}$$

If applied to noisy image sequences, all gradient techniques suffer from the noise which causes inaccuracy of the gradient estimates. To suppress the inaccuracy, regularization methods such as smoothness constraints are often used [18, 69].

Gradient techniques in motion-compensated filtering

Gradient techniques for motion-estimation were for instance used by Huang and Hsu [6] in combination with temporal mean and median filters. Popular descendants of this method are *pel-recursive* motion estimators where the approximation in Equation (3.9) is refined by solving (3.11) recursively [70, 71, 72]. Pel-recursive motion estimation has been used in a number of motion-compensated filtering methods. Among them are Kak filters [52, 53], recursive filters [7, 47] and a Bayesian approach [59].

3.3.3 Matching techniques

A popular motion-estimation technique is that of matching [6]. In this method, a number of unidirectional supports are defined as:

$$[g(i-d_i, j-d_j, k-1),\ g(i,j,k),\ g(i+d_i, j+d_j, k+1)],\quad d_i, d_j \in S. \qquad (3.13)$$

Here, S describes the candidate set, which is typically limited, for instance to: $S = [-2, -1, 0, 1, -2]$. Among the unidirectional supports the one with the smallest signal variance is selected:

$$\hat{d}_i, \hat{d}_j \ \leftarrow \ \min_{d'_i, d'_j} \text{var.}\{g(i-d'_i, j-d'_j, k-1),\ g(i,j,k),\ g(i+d'_j, j+d'_j, k+1)]\}. \quad (3.14)$$

The parameters $\hat{d}_i$ and $\hat{d}_j$ comprise the resulting motion estimate.

An extension of the matching method is block matching, where two-dimensional supports are matched. Then, the image data within a block in the current frame is matched, using a criterion function, to a block in the previous frame. Typical criterion functions used in block-matching algorithms are:

$$\hat{d}_i, \hat{d}_j \ \leftarrow \ \min_{d'_i, d'_j \in S} \sum_{p,q \in A} |g(i-p-d'_i, j-q-d'_j, k-1) - g(i-p, j-q, k)|^c. \quad (3.15)$$

Comprising the square error for $c = 2$ and the absolute difference for $c = 1$. A is the block support and S is the candidate area. A "full-search" algorithm evaluates the criterion function at every location within the candidate area. To decrease the calculational effort, several strategies such as "three-step" and "cross-search" were developed [2]. Practical block matching will only supply one motion vector for each block. Some sort of interpolation scheme is then used to generate a vector for each pixel.

In general, the noise sensitivity of matching methods depends on the block size, search algorithm and criterion. We will consider ways to robustify the popular block-matching algorithm in Section 3.4.2

Matching techniques in motion-compensated filtering

It is straightforward to incorporate the matching method in, for instance, a temporal FIR filtering strategy, because it simultaneously compensates the signal as can be seen from Equation(3.13). This property was exploited in temporal median and average filters [6], and concatenated estimators [54].

The 2D variant, block matching, is quite popular in motion-compensated filtering. Among the filter strategies for which it was used are a spatio-temporal Volterra filter [30], a Bayesian approach [57], a decomposition method [14] and an LLMMSE filter [55].

3.4 Noise-robust motion estimation

The observation noise limits the usefulness of motion estimation in general. This is especially apparent at lower SNRs [13, 64]. In areas with a low image contrast a non-robust motion estimator is easily distracted by the noise.

There are some ways to avoid inaccuracies caused by noise, we consider two. The first method is to *cure* an existing motion estimator, for instance by estimating from pre-filtered frames or by tuning and/or modification. This is done by including noise in the motion model and deriving a noise-robust estimation scheme. The second method, which has appeared recently, is to regard motion estimation and filtering as coupled strategies. That is, the motion and the original sequence are *simultaneously* estimated from the noisy observation using a single criterion function [73].

In this section we first consider several ways to cure a motion estimator from noise sensitivity. As an example, a noise-robust block-matching algorithm that includes some of these remedies is presented in Section 3.4.2. Simultaneous motion estimation and filtering is the subject of Section 3.4.3.

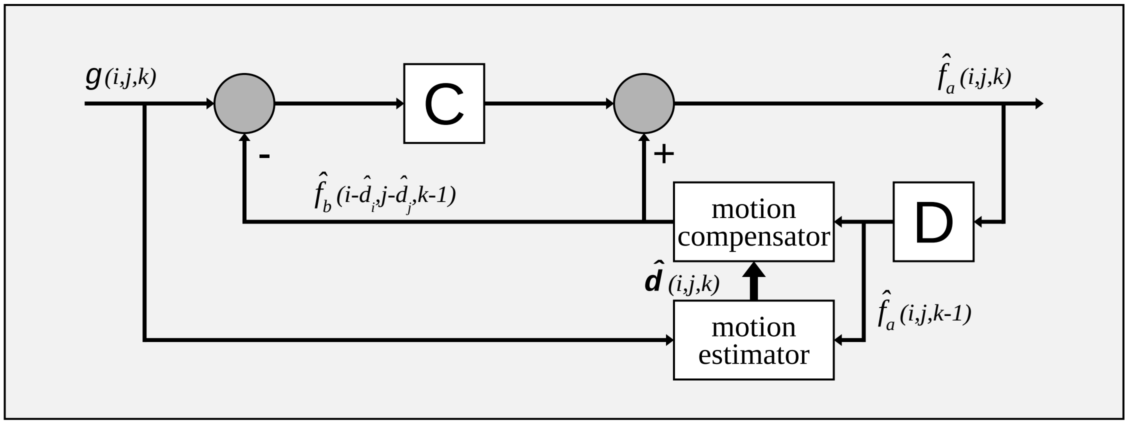


Figure 3.3: Motion estimation using a pre-filtered frame is easily accomplished in the switching filters

3.4.1 Improving the noise robustness of an existing method

The noise sensitivity of an existing method can be decreased by considering the following approaches:

- use pre-filtered frames to estimate motion,
- use an increased amount of data to invoke an averaging effect,
- apply a noise-insensitive criterion function,
- use a tracking algorithm that enforces consistent estimates.

The pre-filtering of frames prior to motion estimation

The usual remedy proposed in the early motion-compensated schemes was to pre-filter the observed frames with a simple spatial filter [6, 52]. An obvious method of pre-filtering is to use the previous filter result. This can easily be accomplished with the switching filters as seen in Figure 3.3 where motion estimation is performed within the feedback loop.

In [43] a spatio-temporal motion-compensated Kalman filter is described where the motion is initially estimated from the filter result on the previous frame and the current observation. After motion-compensated filtering of the current frame using these estimates, an additional motion estimate is performed between the previous and initial filter result. This motion estimate is then used for an additional and also final filtering result.

Using an increased amount of data to invoke an averaging effect

Incorporating more data per motion estimate will decrease the noise sensitivity because of the averaging effect. Ways to increase the amount of data are for instance increasing the matching window in block-matching methods and the derivation of gradient estimates from a larger window in the gradient method. However, using more data will produce inaccurate vector fields because larger spatial areas are more likely to cover more than one object with different motion vectors.

Applying a noise-insensitive criterion function

A useful modification is taking the noise explicitly into account in the criterion function. Some techniques that use a noise-robust criterion function are the method using the Generalized Maximum-Likelihood (GML) criterion by Namazi and Lee [74], and the method using cumulants by Anderson and Giannakkis [75] for Gaussian noise. Another noise-robust criterion function suited for block matching is presented by Kleihorst *et al.* [55, 76]. This is based on triple correlation which is blind to Gaussian noise [77].

Boyce [24] has improved the criterion function of a block-matching scheme by including a-priori knowledge of the noise. She only performs a motion estimate for a certain block if the Mean Absolute Difference (MAD) without motion exceeds a certain threshold which is based on the noise variance. Also, if the displacement found will not result in a convincingly lower MAD, then it is assumed that the difference was caused by noise instead of motion. This simple improvement of a block matcher has been successfully applied with temporal averaging [24], and with a spatio-temporal median filter [38].

Using a tracking algorithm that enforces motion estimate consistency

Enforcing consistency in the motion estimates is done by stimulating similarity between the motion vectors of the entire frame. Motivated by similar assumptions, the regularized estimation algorithms that assume spatial (and/or temporal) smoothness of the vector field [18, 66] were introduced.

Hierarchical motion-estimation algorithms [69, 78] are inherently spatially consistent. In these algorithms, the frames are represented by a resolution pyramid containing the image at various resolution levels. The estimation starts at the image representation that has the lowest resolution. The motion vectors found in this image are used as initial estimates for the estimation process at a lower level with a higher spatial resolution. At this lower level, only fine adjustments of the motion vectors are permitted.

There are some hierarchical motion-estimation algorithms exploited in filtering methods. One is the "Fogel" [69] algorithm which is a gradient-based algorithm with additional smoothness constraints. Another is the hierarchical block-matching algorithm by Bierling [78] which uses regular block matching in every resolution level. The "Fogel" algorithm was used with LLMMSE [46], AWA [8, 27] and Wiener filters [23]. The hierarchical block-matching scheme was used with the spatio-temporal median [38], AWA [8] and spatio-temporal Kalman filters [43].

Recursive motion estimation can also be used to enforce consistency as in the recursive block-matching algorithm by de Haan *et al.* [5, 79, 80]. This algorithm was used in combination with the decomposition method [13, 14, 64] and with an LLMMSE filter in [55]. It will be considered in greater detail in the following.

3.4.2 A noise-robust block-matching algorithm

A typical criterion function used in block matching is the correlation function:

$$\mathrm{Crit}\{\boldsymbol{d}', \boldsymbol{g}\} = E_{i,j}\{g(i - d'_i, j - d'_j, k-1)\; g(i,j,k)\}. \tag{3.16}$$

Here, $E_{i,j}$ denotes averaging over the (finite) block area. A drawback of this function is that it is rather noise sensitive [77].

We propose to modify the correlation function to a *triple*-correlation function in the following way:

$$\mathrm{Crit}\{\boldsymbol{d}', \boldsymbol{g}\} = E_{i,j}\{g(i - d'_i, j - d'_j, k-1)\; g(i,j,k)\; g(i + d'_i, j + d'_j, k+1)\}. \tag{3.17}$$

The advantages of this new criterion function based on 3^{rd} order statistics are twofold. First, the use of three frames (Figure 3.4) will enforce a smooth temporal variation of the motion field. Second, the ratio of signal-induced value and noise-induced value of the triple-correlation measure (3.17) is high because of the 3^{rd}-order terms in $g(i,j,k)$ and $n(i,j,k)$. Note that the 3^{rd}-order moment of zero-mean, symmetrically distributed noise is zero [77, 81].

In comparison to [75] where the use of cumulants was also suggested in a criterion, the use of 3 instead of 2 consecutive frames avoids the use of the "dummy" variables to establish the third term. These dummy variables made the problem computationally very demanding.

The reason why (3.17) is less sensitive to the noise compared to (3.16) can be seen

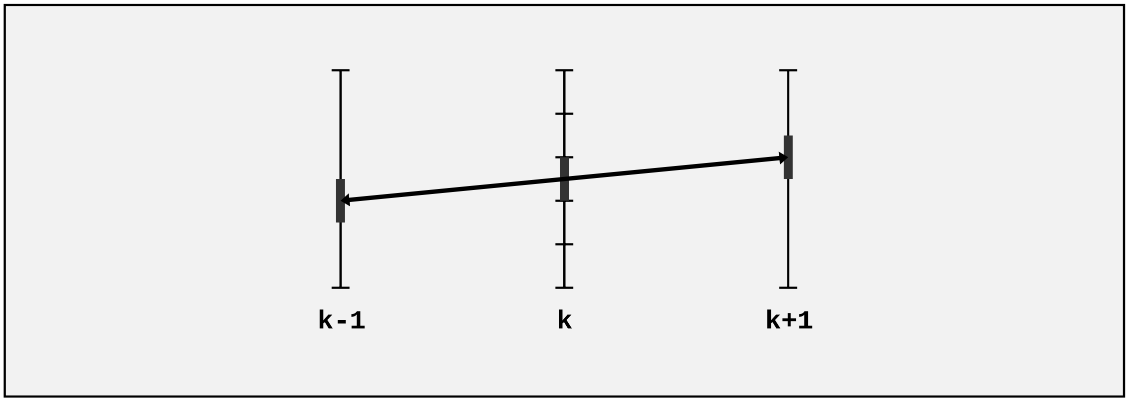


Figure 3.4: By using three frames in the criterion function temporal smoothness is increased and a triple-correlation criterion function can be applied

by inserting the motion model from Equation (3.6) and expanding the product. For this, we use the following short-hand notation:

$$n_{k-1} = n(i - d'_i, j - d'_j, k-1), \quad n_k = n(i,j,k), \quad n_{k+1} = n(i + d'_i, j + d'_j, k+1),$$
$$f_{k-1} = f(i - d'_i, j - d'_j, k-1), \quad f_k = f(i,j,k), \quad f_{k+1} = f(i + d'_i, j + d'_j, k+1).$$

Using $g(i,j,k) = f(i,j,k) + n(i,j,k)$, yields for the *correlation* criterion (3.16):

$$\mathrm{Crit}\{\boldsymbol{d}', \boldsymbol{f} + \boldsymbol{n}\} = E_{i,j}\{f_k f_{k-1}\} + E_{i,j}\{n_k f_{k-1} + n_{k-1} f_k\} + E_{i,j}\{n_k n_{k-1}\}. \quad (3.18)$$

This can be seen as a match on the noise-free data and error terms caused by the noise. The *triple-correlation* criterion (3.17) now becomes:

$$\begin{aligned}\mathrm{Crit}\{\boldsymbol{d}', \boldsymbol{f} + \boldsymbol{n}\} &= E_{i,j}\{f_{k-1} f_k f_{k-1}\} + E_{i,j}\{n_{k-1} f_k f_{k+1} + n_k f_{k-1} f_{k+1} \\ &+ n_{k+1} f_{k-1} f_k + n_k n_{k+1} f_{k-1} + n_{k-1} n_{k+1} f_k + n_{k-1} n_k f_{k+1}\} \\ &+ E_{i,j}\{n_{k-1} n_k n_{k+1}\}. \quad (3.19)\end{aligned}$$

At first, the error term associated with this criterion may seem larger than the error term in (3.18). However, the term regarding the match on the noise-free intensity data is much larger in value, creating a higher data-to-error ratio.

In Figure 3.5 the typical shapes of both criterion functions as a function of $\boldsymbol{d}'$ for different noise levels are illustrated. The criterion function (3.17) is more peaked about the true displacement and approaches zero for incorrect displacements. This will give a more accurate estimate.

By enforcing consistency, the block matcher will deviate less from an estimated motion path due to noise. This can be achieved in a block matcher by introducing recursion [79]. A candidate motion vector $\boldsymbol{d}'$ is then established by the sum of a prediction and an update. For the prediction part the result found for a previous block is used. For instance, a recursion from the left with block sizes of $N \times N$ yields:

$$\boldsymbol{d}'(i,j,k) = \hat{\boldsymbol{d}}(i - N, j, k) + \boldsymbol{u}, \quad (3.20)$$

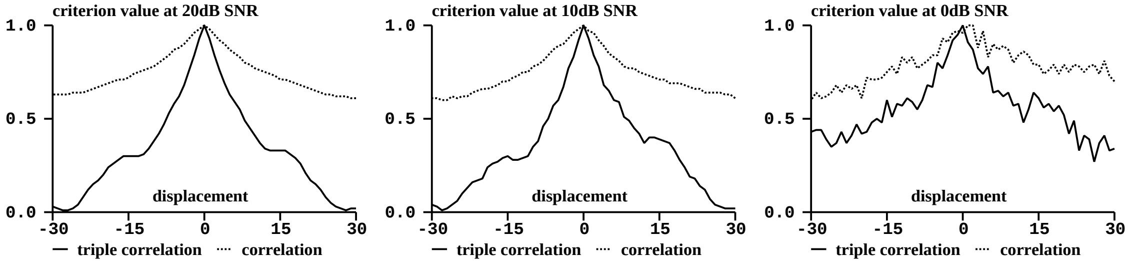


FIGURE 3.5: THE TYPICAL SHAPES OF THE CORRELATION AND TRIPLE-CORRELATION CRITERION FUNCTIONS FOR SNRS OF 20, 10 AND 0 dB

where $\boldsymbol{u}$ is a vector from the update set:

$$\boldsymbol{u} \in \left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ -1 \end{pmatrix}, \begin{pmatrix} -1 \\ 0 \end{pmatrix} \right\}, \tag{3.21}$$

which is restricted in order to stimulate consistency in the estimate.

Using only *one* direction of recursion will cause convergence inaccuracies at motion discontinuities, e.g. moving objects. Because of the restricted update set, the estimate will only slowly converge to the true value. More accurate estimates are obtained using *four* directions of recursion:

$$\begin{aligned} \boldsymbol{d}'(i,j,k) &= \hat{\boldsymbol{d}}(i-N,j,k) + \boldsymbol{u}, \\ \boldsymbol{d}'(i,j,k) &= \hat{\boldsymbol{d}}(i,j-N,k) + \boldsymbol{u}, \\ \boldsymbol{d}'(i,j,k) &= \hat{\boldsymbol{d}}(i,j+N,k-1) + \boldsymbol{u}, \\ \boldsymbol{d}'(i,j,k) &= \hat{\boldsymbol{d}}(i+N,j,k-1) + \boldsymbol{u}. \end{aligned} \tag{3.22}$$

Note that two spatial recursions and two temporal recursions are included. The final estimate becomes that candidate vector which results in the smallest value of the criterion function (3.17). For a more detailed description of the recursive block-matching algorithm the reader is referred to [5, 80].

3.4.3 Simultaneous motion estimation and filtering

The filtering methods that we have considered so far all perform motion estimation and filtering separately. As both displacement and original signal are estimated from the noisy observation, they can be seen as coupled estimation problems [4, 18, 73, 82].

Simultaneous estimation can be stated formally as:

$$\hat{\boldsymbol{f}}, \hat{\boldsymbol{d}} \leftarrow \min_{f',d'} \operatorname{Crit}\{\boldsymbol{g}, \boldsymbol{f}', \boldsymbol{d}'\} \tag{3.23}$$

where the criterion function remains to be chosen.

In general, simultaneous motion estimation and filtering are computationally far more expensive than separate motion estimation and filtering. Chan *et al.* [4] used a maximum-likelihood approach to simultaneously estimate motion and original sequence in low-dosage cine-angiographic image sequences. Rewriting their method for intensity filtering, they considered maximizing the following likelihood function:

$$\hat{\boldsymbol{f}}(k), \hat{\boldsymbol{d}}(k) \;\leftarrow\; \max_{f',d'} \; p\{\boldsymbol{g}(k), \boldsymbol{g}(k-1) | \boldsymbol{f}', \boldsymbol{d}'\}, \tag{3.24}$$

where $\boldsymbol{g}(k), \boldsymbol{f}(k), \boldsymbol{d}(k)$ are the observed and original frame and the vector field between frames k and $k-1$. Chan *et al.* [4] proposed to use the Expectation-Maximization (EM) algorithm which is a method for maximizing log-likelihood functions [15, 83].

Starting with estimates $\hat{\boldsymbol{f}}^s, \hat{\boldsymbol{d}}^s$, the EM algorithm finds the conditional expectation of the log-likelihood of $\boldsymbol{f}, \boldsymbol{d}, \boldsymbol{g}$ given the observed data and current estimates. Within each iteration of the EM algorithm Chan *et al.* [4] decoupled the estimation of original intensity and motion field. This resulted in the following interleaved iteration procedure:

$$\hat{\boldsymbol{f}}^{s+1}(k) \;\leftarrow \max_{f'} \; E\{\log p\{\boldsymbol{f}, \boldsymbol{d}, \boldsymbol{g}\} | \boldsymbol{g}(k), \boldsymbol{g}(k-1), \hat{\boldsymbol{f}}^{s}(k), \hat{\boldsymbol{d}}^{s}(k)\}, \tag{3.25}$$

$$\hat{\boldsymbol{d}}^{s+1}(k) \;\leftarrow \max_{d'} \; E\{\log p\{\boldsymbol{f}, \boldsymbol{d}, \boldsymbol{g}\} | \boldsymbol{g}(k), \boldsymbol{g}(k-1), \hat{\boldsymbol{f}}^{s+1}(k), \hat{\boldsymbol{d}}^{s}(k)\}, \tag{3.26}$$

Brailean and Katsaggelos [73, 82] have proposed a pixel-recursive estimator to solve (3.23). They used Markov random fields to model the displacement and original intensity. These models included “line processes” to adapt the models to discontinuities in the motion field or intensity.

To determine the estimates of the displacement and the intensity fields, Brailean and Katsaggelos proposed to maximize the joint a-posteriori probability density function with respect to the motion field, the original intensity and the corresponding line processes. This resulted in a set of filtering equations consisting of two coupled extended Kalman filters.

Driessen [18] follows a simultaneous approach by minimizing the following concatenated energy function:

$$\hat{\boldsymbol{f}}(k), \hat{\boldsymbol{d}}(k) \;\leftarrow\; \min_{f',d'} \; U\{\boldsymbol{f}', \hat{\boldsymbol{f}}(k-1), \boldsymbol{d}', \hat{\boldsymbol{d}}(k-1), \boldsymbol{g}(k)\}. \tag{3.27}$$

Here $U\{\}$ is composed of the following parts:

$$U\{\} = U_g\{\boldsymbol{f}', \boldsymbol{g}(k)\} + U_f\{\boldsymbol{f}', \hat{\boldsymbol{f}}(k-1), \boldsymbol{d}'\} + U_d\{\boldsymbol{d}', \hat{\boldsymbol{d}}(k-1)\}, \tag{3.28}$$

where the operator U_g reflects the observation model, U_f the motion-compensated spatio-temporal model of the original image and U_d a spatio-temporal motion-smoothness constraint. The problem was attended to by an interleaved iteration process:

$$\begin{aligned} \hat{\boldsymbol{f}}^{s+1}(k) &\leftarrow \min_{f'} U\{\boldsymbol{f}', \hat{\boldsymbol{d}}^{s}(k)\}, && (3.29) \\ \hat{\boldsymbol{d}}^{s+1}(k) &\leftarrow \min_{d'} U\{\hat{\boldsymbol{f}}^{s+1}(k), \boldsymbol{d}'\}. && (3.30) \end{aligned}$$

The minimization was performed by sub-optimal Kalman estimators.

3.5 Experimental evaluation

3.5.1 Demonstration of the noise-robust block matcher

In this section we compare the proposed robust recursive block-matching algorithm with a regular full-search block matcher as a function of the SNR. Both algorithms operate with block dimensions of 16×16.

In the upper row of Figure 3.6 a frame from the synthetic image sequence used is shown at noise levels of SNR$=\infty, 20, 10$ and 0 dB. The underlying sequence contains synthetic movement: the background slightly pans and the disk moves along the diagonal. In the center row of Figure 3.6 the motion field found by the full-search block-matching algorithm is shown for the various noise levels. It can be seen that the motion estimator starts breaking down at 10 dB and produces useless results for 0dB. The motion field estimated by the robust block-matching algorithm is shown in the bottom row of Figure 3.6. The results show that the estimator produces reasonable results, even at 0dB SNR.

Figure 3.7 illustrates the motion in the "mobile" and "Trevor" sequences that are used in the experimental evaluation of motion-compensated noise filters. Here, the motion vectors, estimated with block sizes of 16×16 pixels by the noise-robust block-matching algorithm from Section 3.4.2, are shown in overlay with the original images.

3.5.2 Evaluation of motion-compensated filters

In this section we present the experimental results obtained with some representative motion-compensated filtering methods proposed in literature. As original sequences, we have used parts of the "mobile" and "Trevor" sequence. The type of

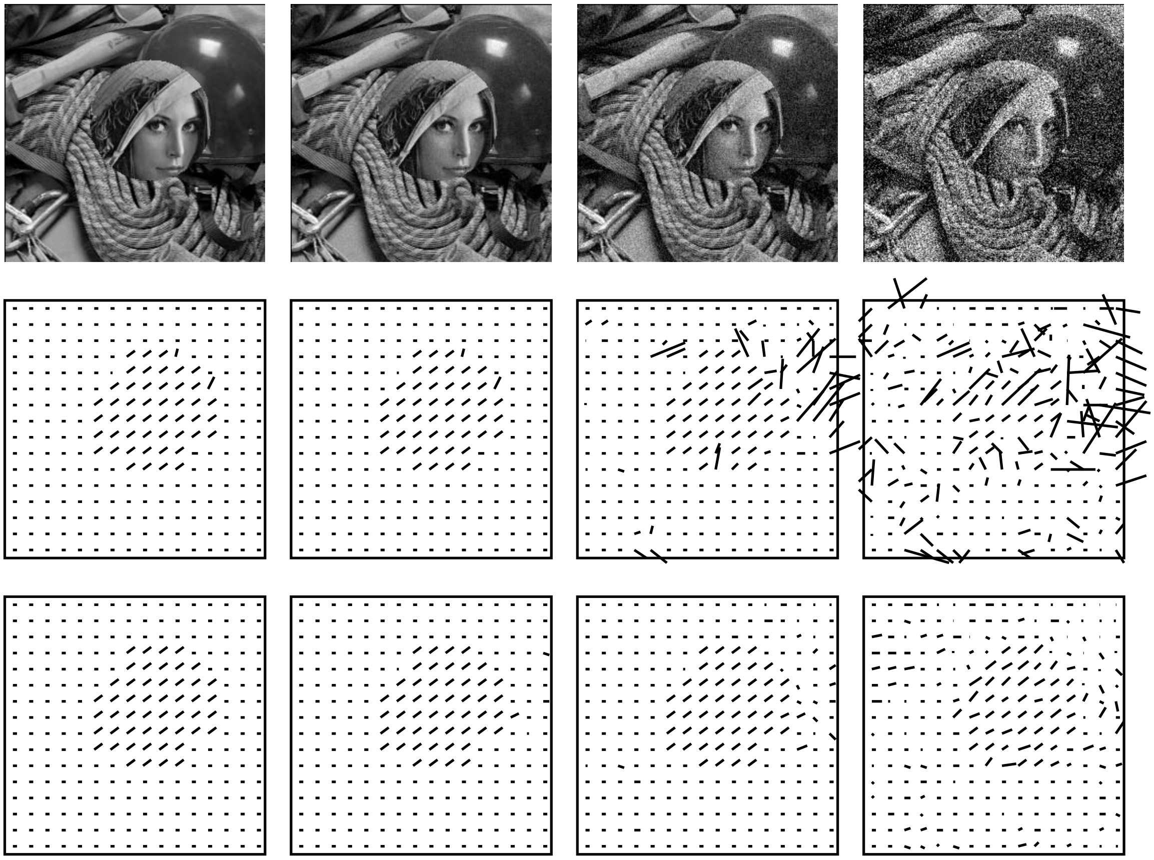


Figure 3.6: In the top row, images from a sequence with synthetic motion at SNRs of ∞, 20, 10 and 0 dB. In the center row results of a full-search block-matching algorithm and on the lower row the results of the noise-robust block-matching algorithm.

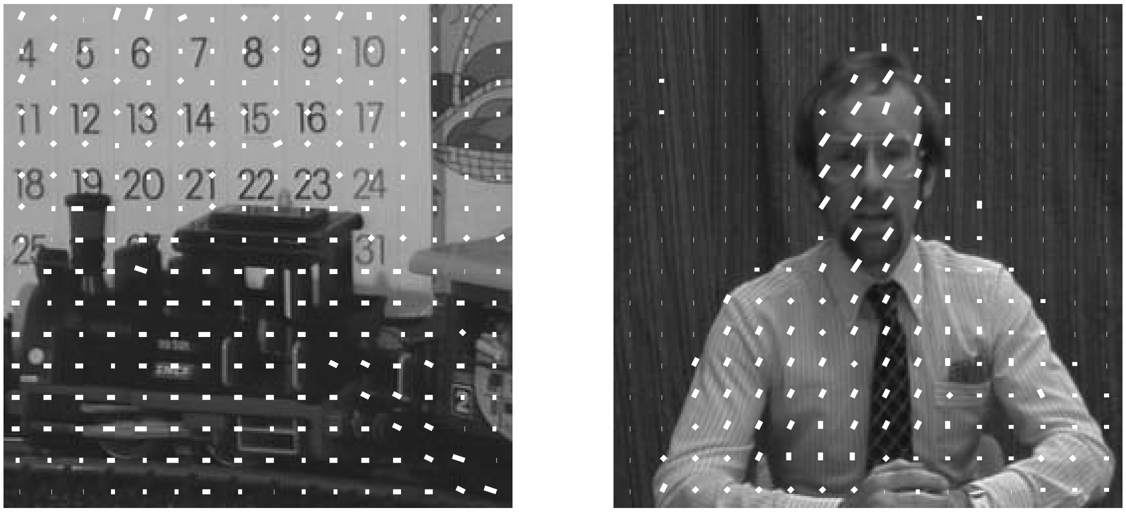


Figure 3.7: The motion vector fields for the "mobile" and "Trevor" sequence

Table 3.1: Summary of the filters evaluated in this chapter

Filter	Motion estimation	Equation	Reference	Description
AWA	BM	(2.5)	[27]	Temporal weighed-averaging filter
Sezan	BM	(2.23)	[46]	Temporal least-squares filter
Dubois	PR	(2.15)	[7]	Recursive switching filter
3D MC Kak	PR	(2.18)	[52]	Recursive 3D least-squares filter
3D sep Kak	PR	(2.18)	[52]	Recursive separable 3D least-squares filter
Adapt	PR	(2.19)	[53]	concatenated 1-D adaptive Kak-filters
3-D AWA	BM	(2.5)	[8]	Spatio-temporal weighted-averaging filter
Kleihorst MC	BM	(2.23)	[48]	Spatio-temporal least-squares filter
SDIE	sim	(3.23)	[73]	Simultaneous Bayesian method

motion present in both sequences is quite different. Namely: the "mobile" sequence contains smooth, mostly translational motion which is relatively easy to follow by recursive motion estimators. In contrast, the "Trevor" sequence contains jerky, relatively unpredictable motion. Both sequences were degraded using simulated additive, white Gaussian noise up to SNRs of 20 and 10 dB.

Noise-filtering algorithms using a block-matching algorithm for motion estimation in their original publication were implemented here using the noise-robust block-matching algorithm described in Section 3.4.2. A similar situation holds for the filters implemented using a pel-recursive algorithm. They were implemented using a pel-recursive motion estimator with smoothness regularization (Section 3.3.2).

The various methods, their symbolic names, and the type of motion estimator used are shown in Table 3.1. In this table "BM" means that a block-matching algorithm was used, "PR" a pel-recursive and "sim" denotes that the motion estimation is part of the simultaneous motion estimation and filtering algorithm. The filter methods "3D MC Kak", "3D sep Kak", "Adapt" and "SDIE" were evaluated at the Digital Signal and Image Processing Laboratory from Northwestern University, Illinois.

The experimental results are summarized in Tables 3.2 and 3.3. The tables contain a number of filter structures that were discussed without motion compensation in the previous chapter. Note that two of these methods were already evaluated without motion compensation in Chapter 2, namely the "3D MC Kak" (a 3-dimensional Kak filter with fixed coefficients) and "Kleihorst MC" (an adaptive switching filter). The SNRi of both filters has improved by including motion compensation. The 3D Kak-filter has a larger improvement for the "mobile" sequence, which contains relatively easy motion for the PR motion estimator involved.

Both a temporal and a spatio-temporal version of the "AWA" filter, an adaptive weighing-averaging filter, are evaluated. The spatio-temporal version has the highest noise suppression for 10 and 20dB SNR. This is because the spatio-temporal version involves more data in the estimation than the temporal version. This is not true for the 20dB corrupted "Trevor" sequence, where the result of the temporal filter is slightly better than the result of the spatio-temporal version. A probable explanation for this behavior is that the motion compensation for this sequence is not perfect, and the larger spatio-temporal support leads to more deterioration of the original signal. The results per frame of the various filters on the distorted sequences are shown in Figure 3.8, Figure 3.9 and Figure 3.10.

It can be seen that the adaptive filters have the highest improvement for the lower SNRs. For higher SNRs, the adaptability is needed less, because the reliability of the motion estimator increases. Especially the simultaneous approach, "SDIE" and the adaptive switching filter "Kleihorst MC" have very good results. The result of these two methods are superior, also from a perceptual point of view. The edges remain sharp and the noise is considerably suppressed.

From the results we see that the overall "best" filters are the adaptive filters. This was also found in Chapter 2. It supports our discussion that motion compensation is not perfect and will therefore not remove all non-stationarities from the temporal signal. In order to achieve good filtering results, adaptation is still necessary.

Table 3.2: Average *improvement in* SNR for the "mobile" sequence.

Filter	*Average Improvement in* SNR (range)	
	SNR=10dB	SNR=20dB
AWA	4.6 (0.2)	3.1 (0.2)
Sezan	4.2 (0.3)	2.6 (0.2)
Dubois	3.3 (0.0)	0.9 (0.8)
3D MC Kak	5.5 (0.4)	1.5 (0.1)
3D sep Kak	6.0 (0.4)	2.2 (0.2)
Adapt	6.2 (0.6)	1.6 (0.4)
3D AWA	7.3 (0.1)	4.1 (0.1)
Kleihorst MC	7.6 (0.2)	4.0 (0.1)
SDIE	7.7 (0.7)	4.4 (0.2)

Table 3.3: Average *improvement in* SNR for the "Trevor" sequence.

Filter	*Average Improvement in* SNR (range)	
	SNR=10dB	SNR=20dB
AWA	4.8 (0.5)	2.9 (0.7)
Sezan	4.6 (0.7)	2.6 (0.6)
Dubois	4.7 (2.6)	1.2 (0.4)
3D MC Kak	5.7 (0.6)	2.1 (0.6)
3D sep Kak	6.4 (0.9)	2.5 (0.4)
Adapt	6.2 (0.6)	2.2 (1.2)
3D AWA	6.9 (0.5)	2.6 (0.5)
Kleihorst MC	7.3 (0.5)	2.4 (0.5)
SDIE	7.7 (1.1)	4.0 (0.5)

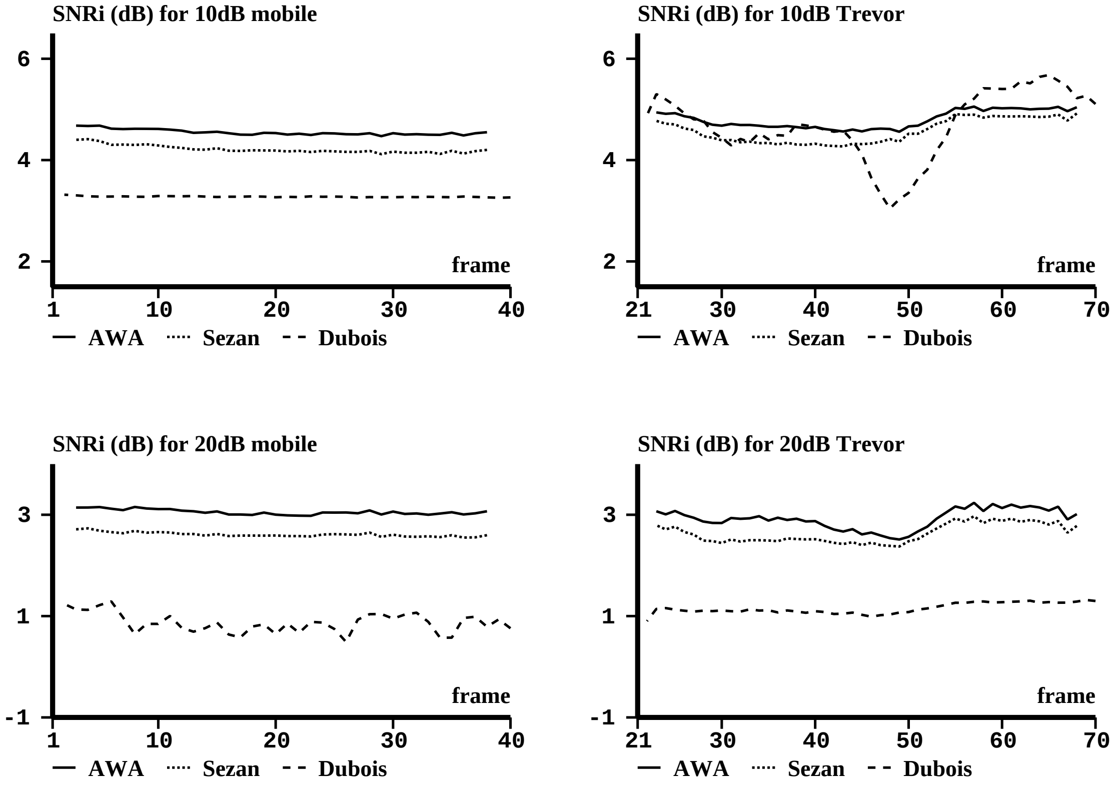


Figure 3.8: Experimental results for the motion-compensated temporal filters.

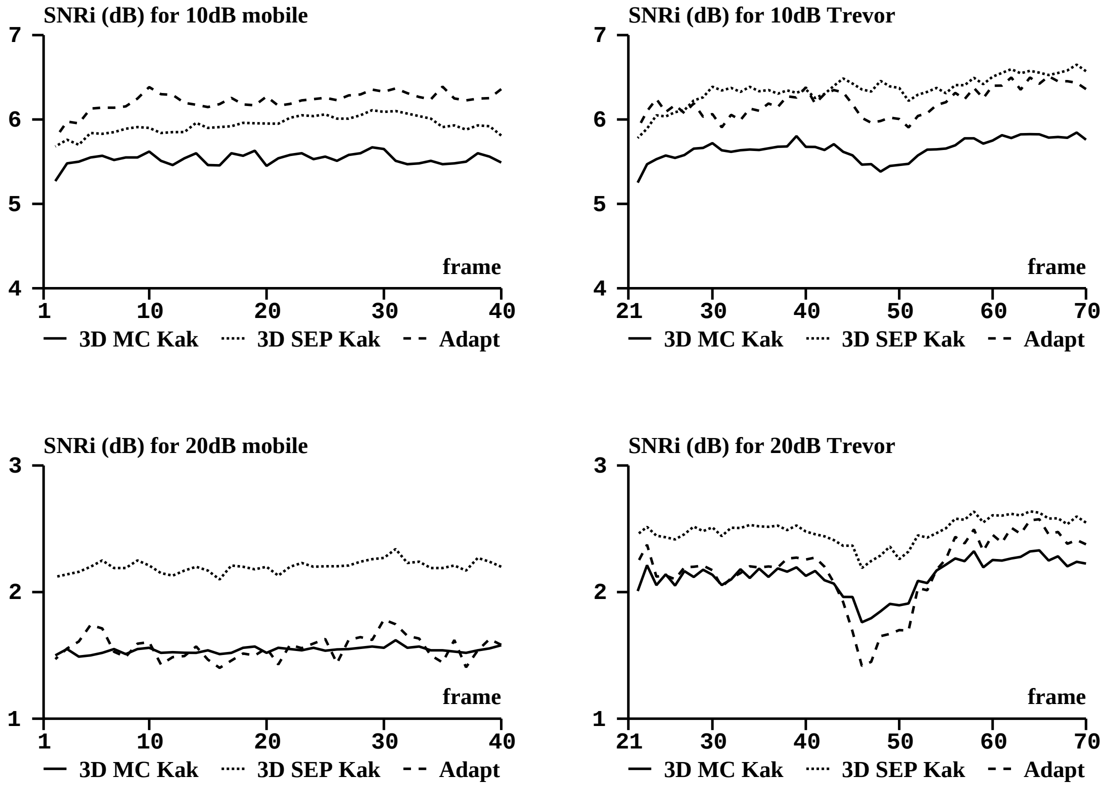


Figure 3.9: Experimental results for the motion-compensated spatio-temporal filters.

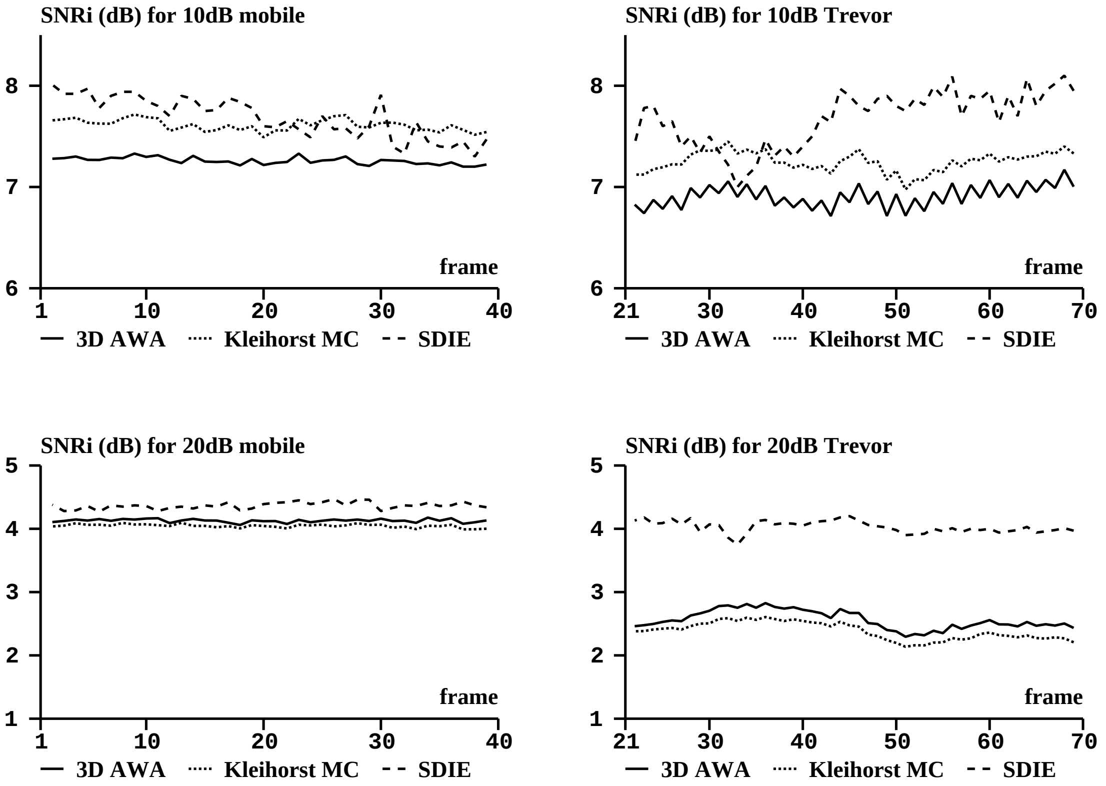


Figure 3.10: Experimental results for the motion-compensated spatio-temporal filters.

Chapter 4

Decomposition and filtering

4.1 Introduction

In the previous chapters we have seen that the main concern in image sequence filtering is handling the non-stationary signals. As the temporal non-stationarities are often caused by motion, an obvious choice for creating stationary temporal signals is using motion compensation. However, the motion models used are never complete and motion estimation is relatively hard in noisy sequences. Therefore, even motion-compensated temporal signals still contain non-stationarities. For this reason, the majority of filters proposed for image sequence noise-filtering are adaptive.

In this chapter, we consider signal decomposition by trend-removal and normalization as a method to transform non-stationary signals into homogeneous variants. In effect, the signal is decomposed into a noise-free non-stationary part and a noisy homogeneous part which is filtered. Signal decomposition is a classical method [84] which is used in time series and image processing [11, 85, 86]. We have proposed a novel image-sequence filtering method by applying it to image sequences. It will be shown that decomposition is beneficial with motion compensated and non-compensated noise filters.

In Section 4.2 we will look from a formal point of view at creating wide-sense stationary signals and the application to image processing. In Section 4.3 we will derive and analyze estimators for the local statistics which are necessary for signal decomposition. In Section 4.4 the implementation aspects of these estimators are investigated. Finally, in Section 4.5 the decomposition method is experimentally investigated in motion-compensated and non-compensated form for Gaussian and Laplacian noise.

4.2 Signal decomposition and its applications

4.2.1 Normalizing from an algebraic point of view

Non-stationarities manifest themselves as non-constant ensemble signal statistics. In image sequences, the dynamic behavior of the local mean and local co-variance cause the poor results of non-adaptive filters. The introduced artifacts manifest themselves as smoothing of spatio-temporal edges.

Any non-stationary signal, such as the observed noisy sequence $\boldsymbol{g}$ can be transformed into a WSS signal $\boldsymbol{y}$ (fixed local statistics) by:

$$\boldsymbol{y} = \boldsymbol{K}(\boldsymbol{g} - \boldsymbol{\mu}_g), \tag{4.1}$$

where the *trend* $\boldsymbol{\mu}_g = E\{\boldsymbol{g}\}$ is a vector to remove non-stationarity in the mean, and $\boldsymbol{K}$ is a transformation matrix used to normalize the co-variance [85]. It appears that $\boldsymbol{K} = \boldsymbol{L}^{-1}$, where $\boldsymbol{L}$ is derived from a Choleski factorization [87] of the symmetric and positive definite co-variance matrix $\boldsymbol{C}_{gg}$ of $\boldsymbol{g}$:

$$\boldsymbol{C}_{gg} = E\{(\boldsymbol{g} - \boldsymbol{\mu}_g)(\boldsymbol{g} - \boldsymbol{\mu}_g)^T\} = \boldsymbol{L}\boldsymbol{L}^T. \tag{4.2}$$

This factorization is successful, because the co-variance matrix of the transformed signal $\boldsymbol{K}(\boldsymbol{g} - \boldsymbol{\mu}_g)$ (Karhunen-Loève transform) becomes:

$$\boldsymbol{C}_{yy} = E\{\boldsymbol{K}(\boldsymbol{g} - \boldsymbol{\mu}_g)(\boldsymbol{g} - \boldsymbol{\mu}_g)^T\boldsymbol{K}^T\} = \boldsymbol{I}, \tag{4.3}$$

which means that the signal $\boldsymbol{y}$ is uncorrelated.

Because the observation $\boldsymbol{g}$ contains a white noise component, $\boldsymbol{C}_{gg}$ will be of full rank so the lower-triangular matrix $\boldsymbol{L}$ is invertible. The resultant transformation matrix $\boldsymbol{K}$ has full rank, to guarantee that no components of the signal are mapped to the null space of $\boldsymbol{K}$ and hence, left out of $\boldsymbol{y}$.

However, computing $\boldsymbol{L}^{-1}$ is an unrealistic task. It will, especially for image sequences, be very large and the amount of storage needed is significant. In order to be computationally realistic, a simple sub-optimal choice for $\boldsymbol{K}$ is a diagonal matrix, with the reciprocals of the local standard deviation for the diagonal elements:

$$\boldsymbol{K} = \text{diag.}\left[1/\boldsymbol{\sigma}_g\right], \tag{4.4}$$

where:

$$\boldsymbol{\sigma}_g = \sqrt{E\{(\boldsymbol{g} - \boldsymbol{\mu}_g)^2\}}. \tag{4.5}$$

This choice will normalize the local variance in the signal by scaling each sample to a variance of one. The overall process will not result in a wide-sense stationary signal $\boldsymbol{y}$, as the amount of correlation between neighboring samples may vary with time. However, as it is more homogeneous, $\boldsymbol{y}$ will be easier to process.

If accurate estimators for the local statistics are used, both $\boldsymbol{\mu}_g$ and $\boldsymbol{\sigma}_g$ are noise free. They comprise the non-stationary part of the decomposition. The signal $\boldsymbol{y}$ includes the observation noise, which was assumed stationary, and the homogeneous part of the original signal.

4.2.2 Application to non-stationary image data

The technique of removing local mean and local deviation (i.e. a diagonal $\boldsymbol{K}$) is often used for time series processing and image processing. Because the latter application is closely connected to image sequence processing, we will investigate this more closer.

Early attempts at restoring degraded images treated the image as a homogeneous random field and used linear techniques for estimation [9]. However, the filters resulting from these approaches are essentially low-pass filters. Therefore, the images are smoothed by these filters which is especially apparent at the edges.

By recognizing that images are in-homogeneous random fields with space-variant mean and variance, Jeng and Woods [11] have improved the overall estimation quality of a Kalman filter operation. First, in a preprocessing step, they estimated the local average and local deviation. By subtracting the local mean from the observation, a *residual* image was created, and by dividing this by its local deviation a *normalized* image was produced. The residual or normalized image was then filtered by a multiple model reduced-update Kalman filter (RUKF).

They noted that the improvement achieved with this method was better than the improvement by applying the RUKF directly to the observed image. One of their suggestions for improvements was to use a more accurate estimator for the local statistics instead of the "box averages" that they were using. Box averages malfunction when the box covers an object edge.

Park and Lee [86] used a median filter as estimator for the local mean. After subtraction, they employed a Wiener filter for filtering the residual signal. They did not normalize the variance, as they assumed the original data to be a covariance-stationary process. Their method was called a double smoother or a reroughing method (Figure 4.1.) The results of Park and Lee are not very good from a noise-suppression view, but the edges remain sharp. Both effects are due to the use of the median filter. This filter is not optimal in the presence of Gaussian noise, so the estimates will be inaccurate and therefore rather noisy. Any noise that passes

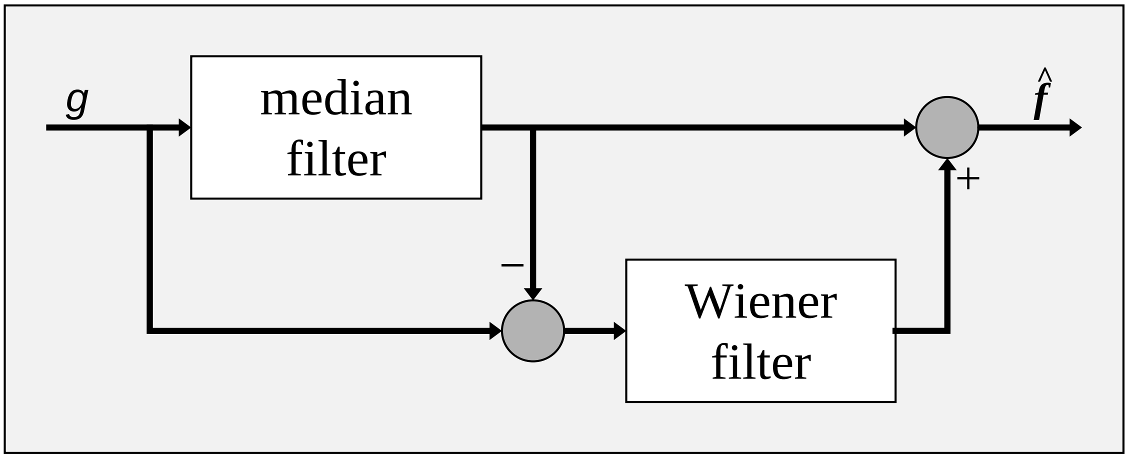


Figure 4.1: Double smoothing using a median and a Wiener filter. The median filter estimates the trend which is removed from the observation and then filtered for noise by the Wiener filter. Trend and filter output are combined to produce the final result.

the estimator is forwarded to the result. The sharp edges are retained because they are roots of the median filter [19, 88].

4.3 Order-statistic estimators

Successful signal normalization strongly depends on the accuracy of the estimators of the local mean $\mu_g(i,j,k)$ and the local deviation $\sigma_g(i,j,k)$. The estimators have to be noise-robust but edge-sensitive. The class of order-statistic estimators has the benefit that it members can be easily designed to be accurate for various noise sources. In addition, the ordering process can be used as a crude form of segmentation (i.e. pixels belonging to the same object are often grouped) in order to deal with edges.

4.3.1 Decomposition and retrieval for image sequences

As seen from Equation (4.1) with $\boldsymbol{K}$ as in (4.4), the simplest form of signal decomposition is to decompose the non-stationary observed signal as: [28, 53]:

$$g(k) = f(k) + n(k) = \mu_g(k) + \sigma_g(k) \cdot y(k). \tag{4.6}$$

Here, $g(k), f(k)$ and $n(k)$ are *temporal* signals (i, j fixed). The slowly changing function $\mu_g(k)$ is known as the *trend*, consisting of the local mean at temporal index k. The local deviation of $g(k)$ is denoted by $\sigma_g(k)$. The normalized temporal signal $y(k)$ has a stationary zero mean and unity deviation. As $\mu_g(k)$ is free of noise, the stationary observation noise is entirely mapped to $y(k)$. The normalized signal can be written using (4.6) as:

$$y(k) = \frac{f(k) + n(k) - \mu_g(k)}{\sigma_g(k)} = \frac{f(k) - \mu_g(k)}{\sigma_g(k)} + \frac{n(k)}{\sigma_g(k)} = r(k) + v(k). \tag{4.7}$$

Here, $v(k)$ is a zero-mean noise signal.

In practice, $y(k)$ is estimated from $g(k)$ in the following way:

$$\hat{y}(k) = \frac{g(k) - \hat{\mu}_g(k)}{\hat{\sigma}_g(k)}, \tag{4.8}$$

where $\hat{\mu}_g(k)$ and $\hat{\sigma}_g(k)$ are estimates of $\mu_g(k)$ and $\sigma_g(k)$ based on the observed noisy sequence.

The homogeneous signal $\hat{y}(k)$, can be filtered using a regular noise filter to estimate $r(k)$. Finally, an estimate of the original image sequence is obtained by combining this estimate $\hat{r}(k)$ with the estimated trend signal and deviation.

$$\hat{f}(k) = \hat{r}(k) \cdot \hat{\sigma}_g(k) + \hat{\mu}_g(k). \tag{4.9}$$

In effect, the decomposition method can be formulated as:

$$\hat{f}(k) = \hat{\mu}_g(k) + \hat{\sigma}_g(k) F \left\{ \frac{g(k) - \hat{\mu}_g(k)}{\hat{\sigma}_g(k)} \right\}, \tag{4.10}$$

where $F\{\}$ is the noise filter operation that estimates $r(k)$ from $y(k)$. This method was mentioned earlier in Section 2.5.

4.3.2 Derivation of the estimators

In order to obtain the normalized signal $y(k)$, $\mu_g(k)$ and $\sigma_g(k)$ have to be estimated from $g(k)$. We will base our approach on the theory of Order-Statistic (OS) estimators. To this end, we first assume that $g(k)$ has the following parametric Probability Density Function (PDF):

$$p(g(k)) = \frac{1}{\sigma_g(k)} q \left\{ \frac{g(k) - \mu_g(k)}{\sigma_g(k)} \right\}, \qquad \sigma_g(k) > 0, \tag{4.11}$$

which we write in shorthand as:

$$g(k) \;\sim\; q\{\mu_g(k), \sigma_g(k)\}. \tag{4.12}$$

Examples of this parametric PDF are the Gaussian, Uniform and Laplacian PDF. Some other PDFs, such as the Poisson and Gamma PDF are not parametric in this way.

The parametric PDF connects to each observation $g(k)$ a parent PDF $q\{\}$, and a particular $\mu_g(k)$ and $\sigma_g(k)$. Let us assume for the moment that the parent PDF $q\{\}$ is known.

Estimators of local statistics based on the class of OS estimators have been successfully applied to digital signal processing [19, 31, 89, 90, 91]. In our application, they have the following generic form:

$$\hat{\boldsymbol{\theta}}(k) = \begin{bmatrix} \hat{\mu}_g(k) \\ \hat{\sigma}_g(k) \end{bmatrix} = \begin{bmatrix} b_{1,1} & b_{1,2} & \dots & b_{1,m} \\ b_{2,1} & b_{2,2} & \dots & b_{2,m} \end{bmatrix} \cdot \begin{bmatrix} g_{(1)}(k) \\ g_{(2)}(k) \\ \vdots \\ g_{(m)}(k) \end{bmatrix} = \boldsymbol{B}\boldsymbol{g}_{()}(k), \tag{4.13}$$

where $\boldsymbol{g}_{()}(k) = [g_{(1)}(k), \dots, g_{(m)}(k)]^T$. The *ranks* $g_{(r)}(k)$ with $g_{(1)}(k) \leq g_{(2)}(k) \leq \dots \leq g_{(m)}(k)$, are ordered realizations of $g(k)$. The subscript "()" denotes that the elements of the vector $\boldsymbol{g}_{()}(k)$ are ordered. Note that (4.13) refers to an ensemble operation; the relation with spatio-temporal filtering will be addressed in Section 4.4. The scalars $b_{p,q}$, with $p \in [1, 2]$, and $q \in [1, 2, \cdots, m]$, are the weights of the OS estimator.

A set of weights $\boldsymbol{B}$ needs to be derived for estimating $\mu_g(k)$ and $\sigma_g(k)$ from $g(k)$. By normalizing $g(k)$ according to (4.8),

$$y(k) = \frac{g(k) - \mu_g(k)}{\sigma_g(k)}, \tag{4.14}$$

$y(k)$ will be distributed as:

$$y(k) \sim q\{0, 1\}. \tag{4.15}$$

Relation (4.14) also holds for the ranks of the observation $g_{(r)}(k)$, $(1 \leq r \leq m)$ and the ranks of $y(k)$, which are denoted by $y_{(r)}(k)$:

$$g_{(r)}(k) = \mu_g(k) + \sigma_g(k) \cdot y_{(r)}(k), \quad 1 \leq r \leq m. \tag{4.16}$$

If we take expectations on both sides of this relation for all r, we arrive at:

$$\begin{aligned} E\{g_{(1)}(k)\} &= \mu_g(k) + \sigma_g(k) \cdot p_{(1)}, \\ E\{g_{(2)}(k)\} &= \mu_g(k) + \sigma_g(k) \cdot p_{(2)}, \\ \vdots \quad & \quad \vdots \\ E\{g_{(m)}(k)\} &= \mu_g(k) + \sigma_g(k) \cdot p_{(m)}. \end{aligned} \tag{4.17}$$

Here, $p_{(r)} = E\{y_{(r)}(k)\}$ are the *rank averages* of $y(k)$. Note that these averages are known because the PDF of $y(k)$, $q(0, 1)$ is fixed. So, if $q()$ is specified, they can be calculated.

Equation (4.17) shows that for ordered realizations of $g(k)$ we expect to find a set of m linear relations, which is over-determined if $m > 2$. The parameters $\hat{\mu}_g(k)$

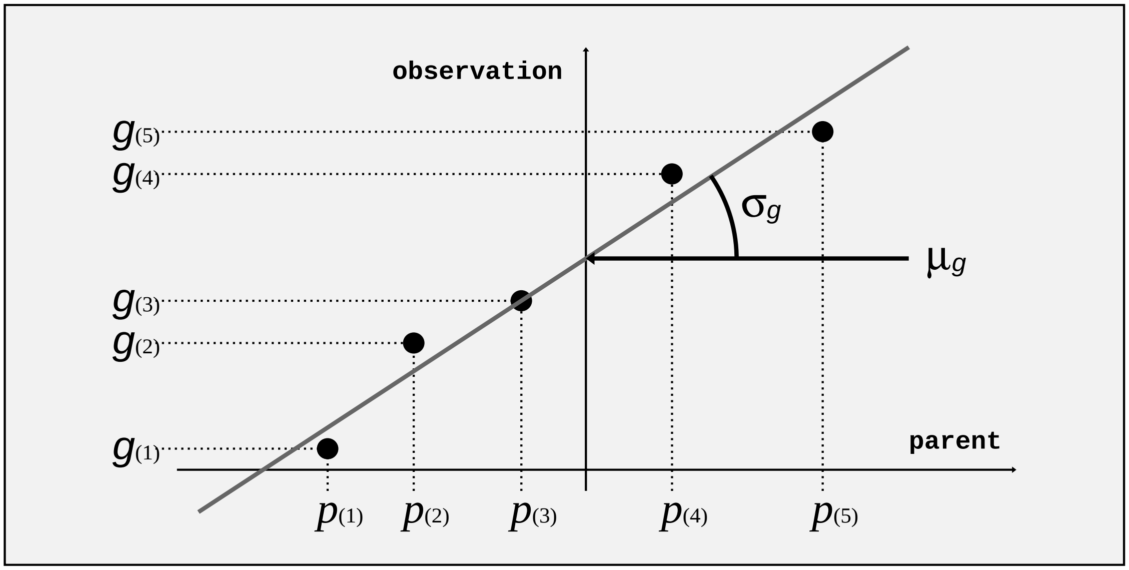


Figure 4.2: A linear relation is expected between the ordered observations on the vertical axis, and the rank averages from the parent PDF on the horizontal axis. The parameters of this relation are the local mean $\mu_g(k)$ and the local deviation $\sigma_g(k)$.

and $\hat{\sigma}_g(k)$, governing the linear relations, can be estimated using linear regression [92], as illustrated in Figure 4.2 for $m = 5$. The offset in the linear relation is $\mu_g(k)$ and the elevation is equal to $\sigma_g(k)$.

To formulate the solution to this regression problem, we first rewrite Equation (4.17) as follows:

$$E\left\{\boldsymbol{g}_{()}(k)\right\} = \begin{bmatrix} 1 & p_{(1)} \\ \vdots & \vdots \\ 1 & p_{(m)} \end{bmatrix} \boldsymbol{\theta}(k) = \begin{bmatrix} \mathbf{1} & \boldsymbol{p}_{(y)} \end{bmatrix} \boldsymbol{\theta}(k) = \boldsymbol{A\theta}(k). \tag{4.18}$$

Where:

$$\boldsymbol{\theta}(k) = \begin{bmatrix} \mu_g(k) \\ \sigma_g(k) \end{bmatrix}. \tag{4.19}$$

In practice, the ensemble averages over $\boldsymbol{g}_{()}(k)$ are not available. Instead, they are approximated by the observations and Equation (4.18) becomes:

$$\boldsymbol{g}_{()}(k) = \boldsymbol{A\theta}(k) + \boldsymbol{\epsilon}(k). \tag{4.20}$$

We assume that $\boldsymbol{\epsilon}(k)$, the residual error caused by dropping the expectation, is zero-mean and uncorrelated with $\boldsymbol{g}_{()}(k)$. The linear relation will most likely not exactly have the same parameters for each rank. Therefore, a measure that finds the best fit of the linear relation in least-squares sense is used. This "generalized least-squares estimate" of $\boldsymbol{\theta}(k)$ is found by minimizing the square of the residual error: $\boldsymbol{\epsilon}^T(k)\boldsymbol{\epsilon}(k)$ for $\boldsymbol{\theta}(k)$:

$$\hat{\boldsymbol{\theta}}(k) \quad \longleftarrow \quad \min_{\boldsymbol{\theta}(k)} \quad (\boldsymbol{g}_{()}(k) - \boldsymbol{A\theta}(k))^T \boldsymbol{C}_{(yy)}^{-1} (\boldsymbol{g}_{()}(k) - \boldsymbol{A\theta}(k)). \tag{4.21}$$

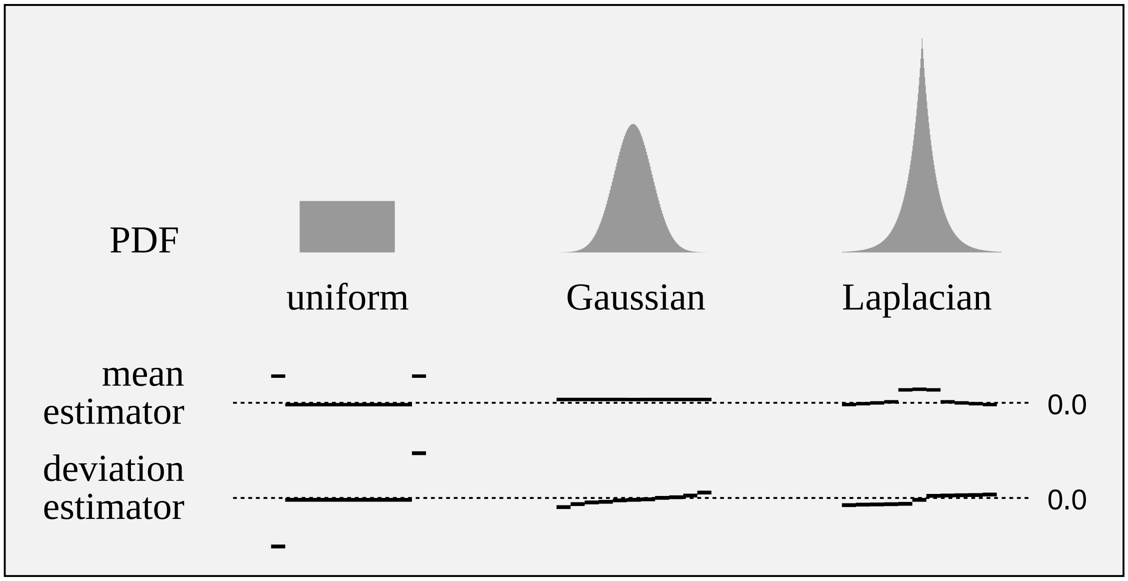


Figure 4.3: The uniform, Gaussian and Laplacian PDFs and their corresponding estimator weights for $\mu_g(k)$ and $\sigma_g(k)$. The exploitation of the (ordered) values is proportional to the probability of the values.

The above inner-product is weighed with the inverse co-variance matrix of $\boldsymbol{y}_{()}(k)$ to remove the correlation introduced by ordering ensemble measurements from $g(k)$. This matrix is composed as:

$$\boldsymbol{C}_{(yy)} = E\left\{\begin{bmatrix} y_{(1)} - p_{(1)} \\ \vdots \\ y_{(m)} - p_{(m)} \end{bmatrix} \begin{bmatrix} y_{(1)} - p_{(1)} & \dots & y_{(m)} - p_{(m)} \end{bmatrix}\right\}. \tag{4.22}$$

The solution to Equation (4.21) is given by:

$$\hat{\boldsymbol{\theta}}(k) = (\boldsymbol{A}^T\boldsymbol{C}_{(yy)}^{-1}\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{C}_{(yy)}^{-1}\boldsymbol{g}_{()}(k) = \boldsymbol{B}\boldsymbol{g}_{()}(k). \tag{4.23}$$

Equation (4.23) comprises the OS estimators for local mean and deviation. The coefficients of $\boldsymbol{B}$ can be interpreted as follows. The upper row supplies the weights to find $\hat{\mu}_g(k)$ and the bottom row supplies the weights to find $\hat{\sigma}_g(k)$. If the PDF is wide, such as the uniform PDF, only the extreme values are used. If the PDF is long-tailed, such as the Laplacian, the center values of $\boldsymbol{g}_{()}(k)$ are relied upon. The OS estimator for mean of a Gaussian PDF is equal to the sample average [31, 93]. In this way all values are equally relied upon. The power of (4.23) is that it gives optimal (in least-squares sense) weights for an OS estimator for the parameters of any parametrizable PDF. The estimator weights for some common PDFs are shown in Figure 4.3.

4.3.3 Estimation bias and variance

The quality of an estimator is reflected by its estimation bias and estimation variance. Generally, the estimation bias $\boldsymbol{\theta}(k) - E\{\hat{\boldsymbol{\theta}}(k)\}$ should be zero, and the estimation variance as low as possible. In the case of a non-scalar estimate, the co-variance of the estimates, which reflects their mutual influence due to errors, should be as low as possible. Bias and variance can be expressed as a function of the estimator weights and statistics of the input signal. In this way, estimators can be compared.

In this subsection, we investigate the bias and estimation variance of the OS estimator from Equation (4.23). Furthermore, we consider the special case of symmetric PDFs for $y(k)$ as this is often assumed in practical filtering situations.

Bias and co-variance of the estimators for general PDFs

The OS estimator for local mean and deviation is unbiased, i.e. has a zero bias. This is a property of the least-squares solution and can be shown using (4.23) and (4.18):

$$E\{\hat{\boldsymbol{\theta}}(k)\} = \boldsymbol{B}E\{\boldsymbol{g}_{()}(k)\} = (\boldsymbol{A}^T\boldsymbol{C}^{-1}_{(yy)}\boldsymbol{A})^T\boldsymbol{A}^T\boldsymbol{C}^{-1}_{(yy)}\boldsymbol{A}\boldsymbol{\theta}(k) = \boldsymbol{\theta}(k). \tag{4.24}$$

The co-variance of the estimator is specified by the following matrix:

$$\boldsymbol{C}_{\theta\theta} = \begin{bmatrix} \text{var.}\{\hat{\mu}_g(k)\} & \text{cov.}\{\hat{\sigma}_g(k), \hat{\mu}_g(k)\} \\ \text{cov.}\{\hat{\sigma}_g(k), \hat{\mu}_g(k)\} & \text{var.}\{\hat{\sigma}_g(k)\} \end{bmatrix}.$$

It can be evaluated as follows:

$$\begin{aligned} \boldsymbol{C}_{\theta\theta} &= E\{(\boldsymbol{A}^T\boldsymbol{C}^{-1}_{(yy)}\boldsymbol{A})^{-1}\boldsymbol{g}_{()}(k)\boldsymbol{g}^T_{()}(k)\boldsymbol{C}^{-1}_{(yy)}\boldsymbol{A}(\boldsymbol{A}^T\boldsymbol{C}^{-1}_{(yy)}\boldsymbol{A})^{-T}\} - \boldsymbol{\theta}(k)\boldsymbol{\theta}^T(k), \\ &= (\boldsymbol{A}^T\boldsymbol{C}^{-1}_{(yy)}\boldsymbol{A})^{-1}\ E\{\boldsymbol{g}_{()}(k)\boldsymbol{g}^T_{()}(k)\}\ \boldsymbol{C}^{-1}_{(yy)}\boldsymbol{A}(\boldsymbol{A}^T\boldsymbol{C}^{-1}_{(yy)}\boldsymbol{A})^{-T} - \boldsymbol{\theta}(k)\boldsymbol{\theta}^T(k), \\ &= (\boldsymbol{A}^T\boldsymbol{C}^{-1}_{(yy)}\boldsymbol{A})^{-1}[\sigma^2_g(k)\boldsymbol{C}_{(yy)} + \boldsymbol{A}\boldsymbol{\theta}(k)\boldsymbol{\theta}^T(k)\boldsymbol{A}^T]\boldsymbol{C}^{-1}_{(yy)}\boldsymbol{A}(\boldsymbol{A}^T\boldsymbol{C}^{-1}_{(yy)}\boldsymbol{A})^{-T} \\ &\qquad -\boldsymbol{\theta}(k)\boldsymbol{\theta}^T(k). \end{aligned}$$

Finally, this will leave:

$$\boldsymbol{C}_{\theta\theta} = \sigma^2_g(k)(\boldsymbol{A}^T\boldsymbol{C}^{-1}_{(yy)}\boldsymbol{A})^{-1}.$$

Or in matrix form as:

$$\boldsymbol{C}_{\theta\theta} = \frac{\sigma^2_g(k)}{|\boldsymbol{A}^T\boldsymbol{C}^{-1}_{(yy)}\boldsymbol{A}|}\begin{bmatrix} \boldsymbol{p}^T_{(y)}\boldsymbol{C}^{-1}_{(yy)}\boldsymbol{p}_{(y)} & -\boldsymbol{1}^T\boldsymbol{C}^{-1}_{(yy)}\boldsymbol{p}_{(y)} \\ -\boldsymbol{p}^T_{(y)}\boldsymbol{C}^{-1}_{(yy)}\boldsymbol{1} & \boldsymbol{1}^T\boldsymbol{C}^{-1}_{(yy)}\boldsymbol{1} \end{bmatrix}, \tag{4.25}$$

indicating that the errors in the local mean and local deviance estimates are correlated for general PDFs.

Co-variance of the estimators for symmetric, zero-mean PDFs

It is interesting to analyze the performance of the estimator for *symmetric*, zero-mean PDFs. In order to do this, we first introduce the $\boldsymbol{J}$ operator, which effectively mirrors the elements of a vector:

$$\boldsymbol{J} = \begin{bmatrix} 0 & 0 & \dots & 0 & 1 \\ 0 & 0 & \dots & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & \dots & 0 & 0 \end{bmatrix}, \tag{4.26}$$

with properties: $\boldsymbol{J} = \boldsymbol{J}^T = \boldsymbol{J}^{-1}, \boldsymbol{J}^T\mathbf{1} = \mathbf{1}$. For a symmetric, zero-mean PDF it is evident that the following holds:

$$\boldsymbol{p}_{(y)} = -\boldsymbol{J}\boldsymbol{p}_{(y)}, \qquad \boldsymbol{C}^{-1}_{(yy)} = \boldsymbol{J}\boldsymbol{C}^{-1}_{(yy)}\boldsymbol{J}. \tag{4.27}$$

If we use these equalities on any non-diagonal element of $\boldsymbol{C}_{\theta\theta}$, for instance:

$$-\mathbf{1}^T\boldsymbol{C}^{-1}_{(yy)}\boldsymbol{p}_{(y)} = \mathbf{1}^T\ \boldsymbol{J}\boldsymbol{C}^{-1}_{(yy)}\boldsymbol{J}\ \boldsymbol{J}\boldsymbol{p}_{(y)} = \mathbf{1}^T\boldsymbol{C}^{-1}_{(yy)}\boldsymbol{p}_{(y)}, \tag{4.28}$$

it appears that they are equal to their own negatives, so must be zero. The co-variance matrix now becomes:

$$\boldsymbol{C}_{\theta\theta} = \begin{bmatrix} \frac{\sigma_g^2}{\mathbf{1}^T\boldsymbol{C}^{-1}_{(yy)}\mathbf{1}} & 0 \\ 0 & \frac{\sigma_g^2}{\boldsymbol{p}^T_{(y)}\boldsymbol{C}^{-1}_{(yy)}\boldsymbol{p}_{(y)}} \end{bmatrix}. \tag{4.29}$$

This shows that for symmetric, zero-mean PDFs there is no mutual correlation in the local mean and local deviance estimate.

Variance of the local mean estimate for symmetric PDFs

Estimators for local mean such as the average and median are commonly used in signal processing. Their estimation variances are well-known [31, 89]. To get a clear idea of the special properties of the OS estimator for symmetric PDFs, we will look closer into the variance of the OS estimate of the local mean. This variance can be read from Equation (4.29) to be:

$$\text{var.}\{\mu_g^2(k)\} = \frac{\sigma_g^2(k)}{\mathbf{1}^T\boldsymbol{C}^{-1}_{(yy)}\mathbf{1}}. \tag{4.30}$$

The denominator can be evaluated as:

$$\mathbf{1}^T\boldsymbol{C}^{-1}_{(yy)}\mathbf{1} = \mathbf{1}^T\boldsymbol{L}^{-T}\boldsymbol{L}^{-1}\mathbf{1} = \boldsymbol{u}^T\boldsymbol{u} = |\boldsymbol{u}|^2. \tag{4.31}$$

Further:

$$\text{var.}\{\mathbf{1}^T \boldsymbol{y}_{()}(k)\} = \mathbf{1}^T \boldsymbol{C}_{(yy)} \mathbf{1} = \mathbf{1}^T \boldsymbol{L}\boldsymbol{L}^T \mathbf{1} = \boldsymbol{q}^T \boldsymbol{q} = |\boldsymbol{q}|^2 = m. \tag{4.32}$$

The relation between $\boldsymbol{u}$ and $\boldsymbol{q}$ is:

$$\boldsymbol{q}^T \boldsymbol{u} = \mathbf{1}^T \boldsymbol{L}\boldsymbol{L}^{-1} \mathbf{1} = \mathbf{1}^T \mathbf{1} = m. \tag{4.33}$$

Using Schwarz's inequality we have:

$$|\boldsymbol{q}|^2 \cdot |\boldsymbol{u}|^2 = m|\boldsymbol{u}|^2 \geq (\boldsymbol{q}^T \boldsymbol{u})^2 = m^2, \tag{4.34}$$

which means that:

$$\text{var.}\{\mu_g^2(k)\} = \frac{\sigma_g^2(k)}{\mathbf{1}^T \boldsymbol{C}_{(yy)}^{-1} \mathbf{1}} \leq \sigma_g^2(k)/m. \tag{4.35}$$

Equality exists for a Gaussian PDF. Then, the maximum-likelihood estimator is the sample average, which can be seen as a member of the class of OS estimators (ordering does not change the result). For all other cases, the OS estimate has a lower variance than the sample average [92].

4.4 Implementation aspects

In this section we consider the implementation aspects of the OS estimator (2.12). After discussing the window definition, a way to establish the parent PDF $q\{\}$ is dealt with. This PDF is described by $\boldsymbol{p}_{(y)}$ and $\boldsymbol{C}_{(yy)}$ which will be estimated from the normalized signal itself. The estimation is performed in a way which enables adaptive OS estimators.

4.4.1 Window definition

The OS estimators were derived on the basis of ensemble statistics. However, these are generally not available. Consequently, we have to assume local ergodicy in image sequences which makes it feasible to replace the ensemble statistics by the local spatio-temporal statistics. This means that the ensemble measurements of $g(k)$ are given by the values within a spatio-temporal window of size m.

To support the assumption of ergodicy, the spatio-temporal window must be compact. We have chosen to incorporate the current pixel in combination with its 6 nearest spatio-temporal neighbors:

$$[g(i,j,k-1),\ g(i,j,k+1),\ g(i,j,k),\ g(i-1,j,k),$$

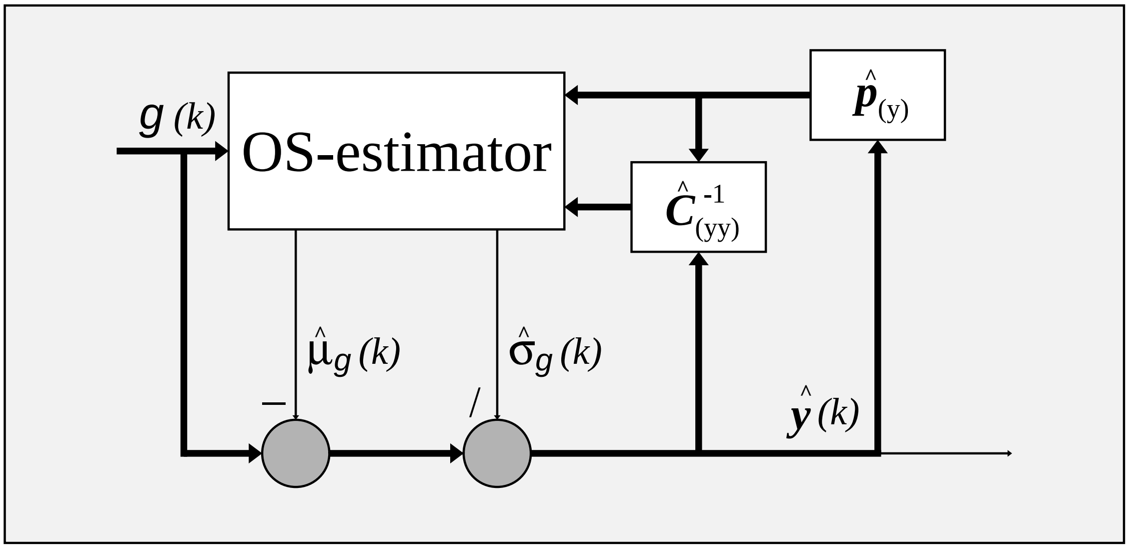


Figure 4.4: The average vector and co-variance matrix of the parent signal, necessary for the estimator, can be established using the normalized signal itself. To support the estimator two update processes, one for the average vector and one for the co-variance matrix are employed.

$$g(i+1, j, k),\ g(i, j-1, k),\ g(i, j+1, k)], \tag{4.36}$$

which proved in a number of experiments to be a good trade-off between low estimation variance and correctness of the assumption of local ergodicity. Other window definitions are discussed in Chapter 5.

4.4.2 Parent PDF

The estimator from Equation (4.23) requires knowledge about the parent PDF, which is expressed by the average vector of an ordered sample from this parent $\boldsymbol{p}_{(y)}$, and the co-variance matrix of this sample, $\boldsymbol{C}_{(yy)}$. In this subsection, we will describe how to estimate them.

The parent PDF can be derived from the normalized signal itself. The vector $\boldsymbol{y}_{()}(k)$ can be used to establish the average vector and the co-variance matrix of the parent PDF, as illustrated in Figure 4.4.

Because it is not possible to accurately describe the co-variance matrix and the average vector based upon a single observation, the accuracy has to be assured by including more data. This is realized using two update processes which are performed for every pixel, jointly with the filtering process. These update processes will also enable tracking of the PDF in non-stationary environments.

4.4.3 Recursively updating the average vector

The vector $\boldsymbol{p}_{(y)}$ can be estimated from previously normalized signals using:

$$\hat{\boldsymbol{p}}_{(y)}(k) = \frac{1}{\sum_{l=1}^{k} \lambda^{k-l}} \sum_{l=1}^{k} \lambda^{k-l} \boldsymbol{y}_{()}(l). \tag{4.37}$$

Here, we have introduced a temporal index with $\boldsymbol{p}_{(y)}$ indicating its changing nature and the updating process. From this, a recurrent equation can be found, so $\hat{\boldsymbol{p}}_{(y)}(k)$ is updated in the temporal direction:

$$\hat{\boldsymbol{p}}_{(y)}(k) = \lambda \hat{\boldsymbol{p}}_{(y)}(k-1) - \frac{\lambda^k \hat{\boldsymbol{p}}_{(y)}(k-1) - \boldsymbol{y}_{()}(k)}{\sum_{l=1}^{k} \lambda^{k-l}}. \tag{4.38}$$

The coefficient λ is a *forgetting factor* [56, 94] which can be assigned values between zero and one. If $\lambda = 1$, all previous samples are weighed equally. This is often referred to as the *pre-windowed* case which is mostly used with stationary data. If $\lambda < 1$, recent observations are weighed more heavily and therefore, the result is more suitable as an estimate of the local statistics of a non-stationary signal. Progressively smaller values of λ compute $\hat{\boldsymbol{p}}_{(y)}(k)$ based upon effectively smaller sections of data. There is an optimal value for a specific signal [95], but this is applicable for off-line processing only. Appropriate practical values for λ with image sequence filtering are $0.97 \cdots 0.99$.

The denominator in (4.38) can be explicitly computed as:

$$\sum_{l=1}^{k} \lambda^{k-l} = \sum_{l=0}^{k-1} \lambda^{l} = \frac{1-\lambda^k}{1-\lambda}, \tag{4.39}$$

or calculated recurrently. The average vector of the observation noise process, scaled to have a unity variance, can be used as the initial $\hat{\boldsymbol{p}}_{(y)}(0)$. In this case, $\hat{\boldsymbol{p}}_{(y)}(k)$ will be optimal for estimating a constant signal immersed in this noise. The average vectors are tabulated for some PDFs in [96].

4.4.4 Recursively updating the inverse covariance matrix

In practical situations, the co-variance matrix $\boldsymbol{C}_{(yy)}$ is approximated at time k by:

$$\hat{\boldsymbol{C}}_{(yy)}(k) = \sum_{l=1}^{k} \lambda^{k-l}\, \tilde{\boldsymbol{y}}_{()}(l) \tilde{\boldsymbol{y}}_{()}^T(l), \tag{4.40}$$

where $\tilde{\boldsymbol{y}}_{()}(l) = \boldsymbol{y}_{()}(l) - \hat{\boldsymbol{p}}_{(y)}(l)$. Again, we have introduced the temporal index k with $\hat{\boldsymbol{C}}_{(yy)}$ to indicate its active nature and the update process. For wide-sense stationary, ergodic data and $\lambda = 1$:

$$\lim_{k\to\infty} \frac{1}{k} \hat{\boldsymbol{C}}_{(yy)}(k) = \boldsymbol{C}_{(yy)}, \tag{4.41}$$

where $\boldsymbol{C}_{(yy)}$ is the true co-variance matrix.

A recurrent expression for (4.40) is:

$$\begin{aligned}\hat{\boldsymbol{C}}_{(yy)}(k) &= \tilde{\boldsymbol{y}}_{()}(k)\tilde{\boldsymbol{y}}^T_{()}(k) + \sum_{l=1}^{k-1} \lambda^{k-l}\ \tilde{\boldsymbol{y}}_{()}(l)\tilde{\boldsymbol{y}}^T_{()}(l),\\ &= \lambda\hat{\boldsymbol{C}}_{(yy)}(k-1) + \tilde{\boldsymbol{y}}_{()}(k)\tilde{\boldsymbol{y}}^T_{()}(k). \qquad (4.42)\end{aligned}$$

In the OS estimator (4.23) we need $\boldsymbol{C}^{-1}_{(yy)}$. To this end, we use the matrix-inversion lemma [56, 94] that states that since the square matrix $\boldsymbol{C}_{(yy)}$ can be written as (4.42), its inverse $\boldsymbol{C}^{-1}_{(yy)}$ may be written as

$$\hat{\boldsymbol{C}}^{-1}_{(yy)}(k) = \frac{\hat{\boldsymbol{C}}^{-1}_{(yy)}(k-1)}{\lambda} - \frac{\frac{\hat{\boldsymbol{C}}^{-1}_{(yy)(k-1)}}{\lambda}\ \tilde{\boldsymbol{y}}_{()}(k)\tilde{\boldsymbol{y}}^T_{()}(k)\frac{\hat{\boldsymbol{C}}^{-1}_{(yy)(k-1)}}{\lambda}}{1+\tilde{\boldsymbol{y}}^T_{()}(k)\frac{\hat{\boldsymbol{C}}^{-1}_{(yy)(k-1)}}{\lambda}\tilde{\boldsymbol{y}}_{()}(k)}, \qquad (4.43)$$

$$= \frac{1}{\lambda}\left[\hat{\boldsymbol{C}}^{-1}_{(yy)}(k-1) - \frac{\hat{\boldsymbol{C}}^{-1}_{(yy)}(k-1)\ \tilde{\boldsymbol{y}}_{()}(k)\tilde{\boldsymbol{y}}^T_{()}(k)\ \hat{\boldsymbol{C}}^{-1}_{(yy)}(k-1)}{\lambda + \tilde{\boldsymbol{y}}_{()}(k)^T\ \hat{\boldsymbol{C}}^{-1}_{(yy)}(k-1)\ \tilde{\boldsymbol{y}}_{()}(k)}\right]. \qquad (4.44)$$

In this recursive algorithm $\hat{\boldsymbol{C}}^{-1}_{(yy)}(0)$ has to be explicitly given. It is often advised to use $\delta\boldsymbol{I}$ for this matrix, with a large δ [56]. As an alternative, a choice more applicable to our situation and in correspondence with $\hat{\boldsymbol{p}}_{(y)}(0)$, is to use the inverse co-variance matrix from an ordered noise sample, with the variance of the noise scaled to unity.

Computational aspects of the update process

A particular situation that is usually detrimental to an update system is when the input is not persistently exciting [94]. This will cause the elements of

$$\hat{\boldsymbol{C}}^{-1}_{(yy)}(k-1)\ \tilde{\boldsymbol{y}}_{()}(k)\tilde{\boldsymbol{y}}^T_{()}(k)\hat{\boldsymbol{C}}^{-1}_{(yy)}(k-1),$$

and

$$\tilde{\boldsymbol{y}}^T_{()}(k)\ \hat{\boldsymbol{C}}^{-1}_{(yy)}(k-1)\ \tilde{\boldsymbol{y}}_{()}(k),$$

to be zero, because $\tilde{\boldsymbol{y}}_{()}(k)$ will finally lie in the null-space of $\hat{\boldsymbol{C}}^{-1}_{(yy)}(k)$. As a result, (4.44) degenerates to:

$$\hat{\boldsymbol{C}}^{-1}_{(yy)}(k) = \frac{\hat{\boldsymbol{C}}^{-1}_{(yy)}(k-1)}{\lambda}.$$

If $\lambda = 1$, the co-variance matrix will not change, but if $\lambda < 1$, the elements of the co-variance matrix start to grow and numerical overflow can occur. It is, however, not directly necessary to correct $\hat{\boldsymbol{C}}^{-1}_{(yy)}(k)$ for this growing. In the estimator (4.23), any multiplication factor in $\hat{\boldsymbol{C}}^{-1}_{(yy)}(k)$ is irrelevant, because this scalar will occur both in direct and reciprocal form.

The recurrent calculation of $\hat{\boldsymbol{C}}^{-1}_{(yy)}(k)$ according to (4.44) is computationally more efficient than using (4.42) and inverting the result directly. Comparing a direct method, which takes at least m^3 multiplications for the inversion (Gauss-Jordan) and an extra m for the update, with the recursive scheme, which takes $3m^2 + 2m$ multiplications, shows that the recursive scheme is preferable for $m > 3$. In the case where $m = 3$ (a relatively small operating window), we have successfully applied Singular Value Decomposition (SVD) to derive $\hat{\boldsymbol{C}}^{-1}_{(yy)}(k)$ [19, 28].

4.4.5 Parent mismatch

The parent PDF can have several shapes. Two global situations can be encountered. First, filtering in a *passive* area, where no motion occurs and no edges are crossed. Second, filtering in an *active* area where several objects are crossed.

In a passive area, the estimation process is equal to estimating a constant signal immersed in noise. Here, the parent PDF is a normalized version of the noise PDF. The local mean is equal to the value of the constant signal (for zero-mean noise), and the local deviation is equivalent to the standard deviation of the noise.

In an active area, with moving objects, the optimal parent PDF becomes less well-defined. To enable accurate estimation of the local mean and deviation, the parent PDF has to be a normalized version of the PDF of a signal including noise and a component generated by moving objects.

Using an update process as considered in the previous subsections for the description of the parent can only guarantee *slow* adaptation to the current (desired) PDF. Slowly tracking the parent PDF unavoidably leads to a mismatch between the actual and estimated PDF. This might lead to biased estimates with a higher estimation variance. Here, we will investigate the effect of a mismatch on the estimation bias.

Estimate bias as a result of parent mismatch

As seen before, the expectation of $\hat{\boldsymbol{\theta}}(k)$ based on the current estimates $\hat{\boldsymbol{A}}$, and $\hat{\boldsymbol{C}}^{-1}_{(yy)}$, is given by:

$$E\{\hat{\boldsymbol{\theta}}(k)\} = (\hat{\boldsymbol{A}}^T \hat{\boldsymbol{C}}^{-1}_{(yy)} \hat{\boldsymbol{A}})^{-1} \hat{\boldsymbol{A}}^T \hat{\boldsymbol{C}}^{-1}_{(yy)} E\{\boldsymbol{g}_{()}(k)\}. \tag{4.45}$$

If $\boldsymbol{g}_{()}(k)$ comes from *another* PDF with local mean $\tilde{\mu}_g(k)$ and local deviation $\tilde{\sigma}_g(k)$, then:

$$E\{\boldsymbol{g}_{()}(k)\} = \left[\begin{array}{cc} \mathbf{1} & \tilde{\boldsymbol{p}}_{(y)} \end{array} \right] \left[\begin{array}{c} \tilde{\mu}_g(k) \\ \tilde{\sigma}_g(k) \end{array} \right] = \tilde{\boldsymbol{A}} \tilde{\boldsymbol{\theta}}(k), \tag{4.46}$$

where $\tilde{\boldsymbol{A}}$ and $\tilde{\boldsymbol{\theta}}(k)$ are based on the actual PDF of $\boldsymbol{g}_{()}(k)$. The bias on $\hat{\boldsymbol{\theta}}(k)$ can be evaluated as follows:

$$E\{\hat{\boldsymbol{\theta}}(k)\} = (\hat{\boldsymbol{A}}^T \hat{\boldsymbol{C}}^{-1}_{(yy)} \hat{\boldsymbol{A}})^{-1} \hat{\boldsymbol{A}}^T \hat{\boldsymbol{C}}^{-1}_{(yy)} \tilde{\boldsymbol{A}} \tilde{\boldsymbol{\theta}}(k). \tag{4.47}$$

For symmetric PDFs this degenerates to:

$$\begin{aligned} \left[\begin{array}{c} E\{\hat{\mu}_g(k)\} \\ E\{\hat{\sigma}_g(k)\} \end{array} \right] &= \left[\begin{array}{cc} \frac{1}{\mathbf{1}^T \hat{\boldsymbol{C}}^{-1}_{(yy)} \mathbf{1}} & 0 \\ 0 & \frac{1}{\hat{\boldsymbol{p}}^T_{(y)} \hat{\boldsymbol{C}}^{-1}_{(yy)} \hat{\boldsymbol{p}}_{(y)}} \end{array} \right] \\ &\cdot \left[\begin{array}{cc} \mathbf{1}^T \hat{\boldsymbol{C}}^{-1}_{(yy)} \mathbf{1} & 0 \\ 0 & \hat{\boldsymbol{p}}^T_{(y)} \hat{\boldsymbol{C}}^{-1}_{(yy)} \tilde{\boldsymbol{p}}_{(y)} \end{array} \right] \left[\begin{array}{c} \tilde{\mu}_g(k) \\ \tilde{\sigma}_g(k) \end{array} \right], \\ &= \left[\begin{array}{cc} 1 & 0 \\ 0 & \frac{\hat{\boldsymbol{p}}^T_{(y)} \hat{\boldsymbol{C}}^{-1}_{(yy)} \hat{\boldsymbol{p}}_{(y)}}{\hat{\boldsymbol{p}}^T_{(y)} \hat{\boldsymbol{C}}^{-1}_{(yy)} \tilde{\boldsymbol{p}}_{(y)}} \end{array} \right] \left[\begin{array}{c} \tilde{\mu}_g(k) \\ \tilde{\sigma}_g(k) \end{array} \right]. \end{aligned} \tag{4.48}$$

It appears that there is no bias in the estimate for local mean and an uncorrelated bias in the estimate for local deviation.

4.5 Experimental evaluation

In this section the proposed algorithm, shown in Figure 4.5, is evaluated on noisy image sequences. We test its performance for several noise levels comprising Gaussian and Laplacian noise. In addition, we test the effectiveness of an optional motion-compensation preprocessing step.

We have again used parts of the "Trevor" sequence (images 21 to 50) and the "mobile" sequence (images 1 to 40). From the original sequences spatial parts of 256 × 256 pixels where selected for processing. For the optional motion estimator we have used the recursive block-matching algorithm which was described in Section 3.4.2 with correlation as the matching criterion (3.16).

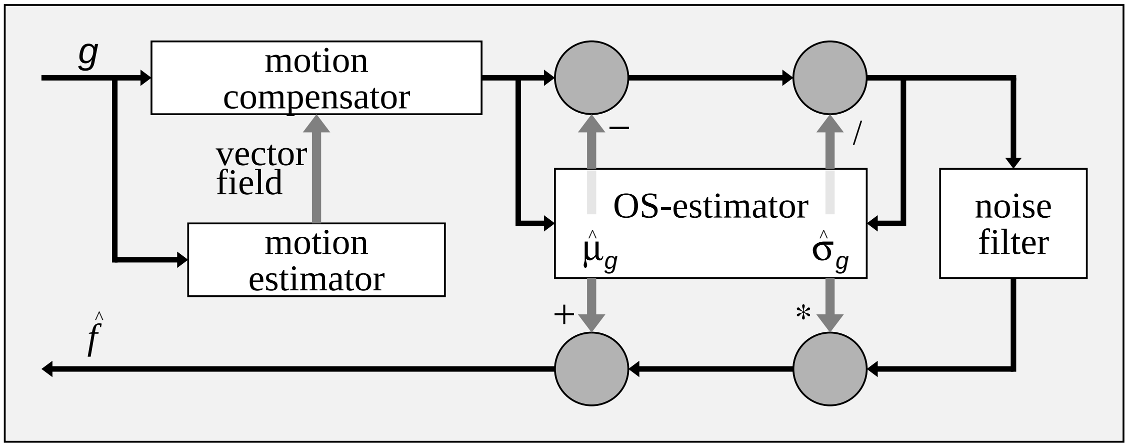


Figure 4.5: An overview of the entire filter. First, in an optional motion-compensation stage, most of the non-stationarities caused by motion can be removed. Then, from this signal the local mean and local deviation are estimated and the signal is normalized. This noisy signal is then filtered by a regular noise filter. The final result is achieved after transforming the filter output.

Table 4.1: Average *improvement in* SNR for the "Trevor" sequence.

Method	*Average Improvement in* SNR (range)		
	SNR=0dB	SNR=10dB	SNR=20dB
Decomposition	8.0 (0.1)	6.3 (0.4)	1.6 (1.1)
MC+Decomposition	7.5 (0.1)	6.4 (0.3)	2.5 (0.9)

The noise filter used in Equation (4.10) is the RLS-filter from Equation (2.26). For both update processes a forgetting factor of $\lambda = 0.97$ was used. This value was derived by experimental tuning for best result given both image sequences at various noise levels. The estimated local deviance was also used to relax the update process of the RLS-filter [19]. This is to reflect the dynamic nature of the variance of the noise $v(k)$ in the normalized signal $y(k)$ as seen in Equation (4.7).

4.5.1 The effects of motion compensation

First, the effect of using motion compensation is illustrated for different amounts of Gaussian noise. For each noise level, the filtering is performed with and without motion compensation. In Table 4.1 the average SNR improvements are given. Compared with the results from Chapters 2 and 3 it can be seen that the decomposition method gives good results. The use of motion estimation has a positive effect on the final result, except for 0dB SNR where the motion estimator fails because of noise. In Figure 4.6, the SNRi curves are shown for 20, 10 and 0dB Gaussian noise, respectively. The precise effects of motion compensation are clearly visible from

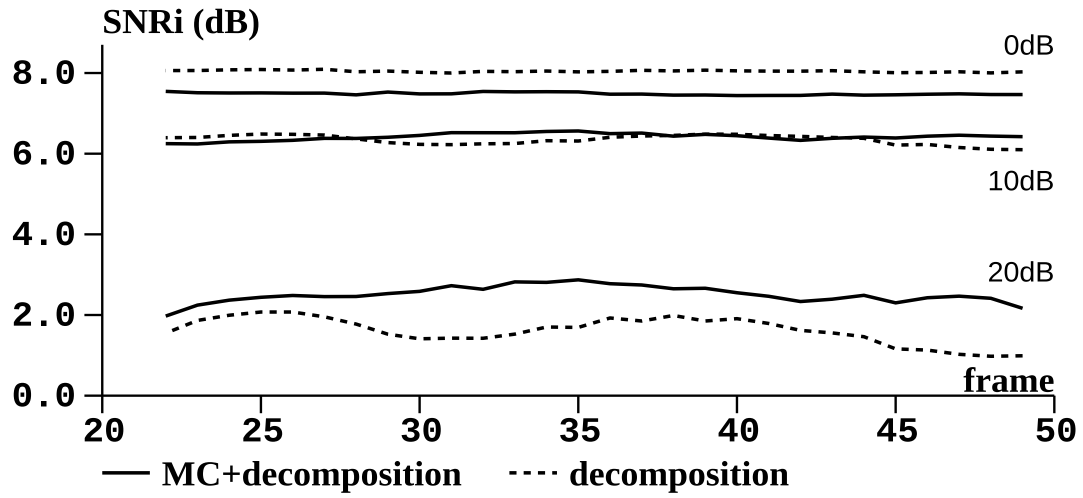


Figure 4.6: Motion compensated and non motion-compensated results of the decomposition method for 0, 10 and 20 dB SNR on the "Trevor" sequence

these curves. For high SNRs (20dB), motion compensation is a useful preprocessing step. For moderate levels (10dB) the two curves are approximately identical, while for low SNRs (0dB) the motion estimation fails. Therefore, the overall noise filtering becomes less successful.

To illustrate the signals created by the OS estimator, we show the estimated local mean and the associated normalized signal for a single frame in Figure 4.7. These frames are taken from the experiment on 10dB data *without* motion compensation. It can be seen that the local mean signal is essentially noise-free and smooth. The normalized signal is stretched and an offset is added for maximum visibility.

The local mean and normalized signal for the experiment *with* motion compensation are shown in Figure 4.8. It can be seen that the motion-compensated local mean signal is sharper and the normalized signal contains fewer spatio-temporal edges. The residual signal is scaled with the same parameters as used for the residual signal in Figure 4.7. The final results of applying the proposed algorithm without and with motion compensation are shown in Figure 4.9 for an SNR of 10dB.

4.5.2 Application to Laplacian noise

We have corrupted the "mobile" sequence with Gaussian and Laplacian noise at a level of 10dB. This resulted in noise variances of $\sigma_n^2 = 320$. The results of this experiment are presented in Figure 4.10. Motion compensation was used in the algorithm. An observed and filtered image of the sequence corrupted with

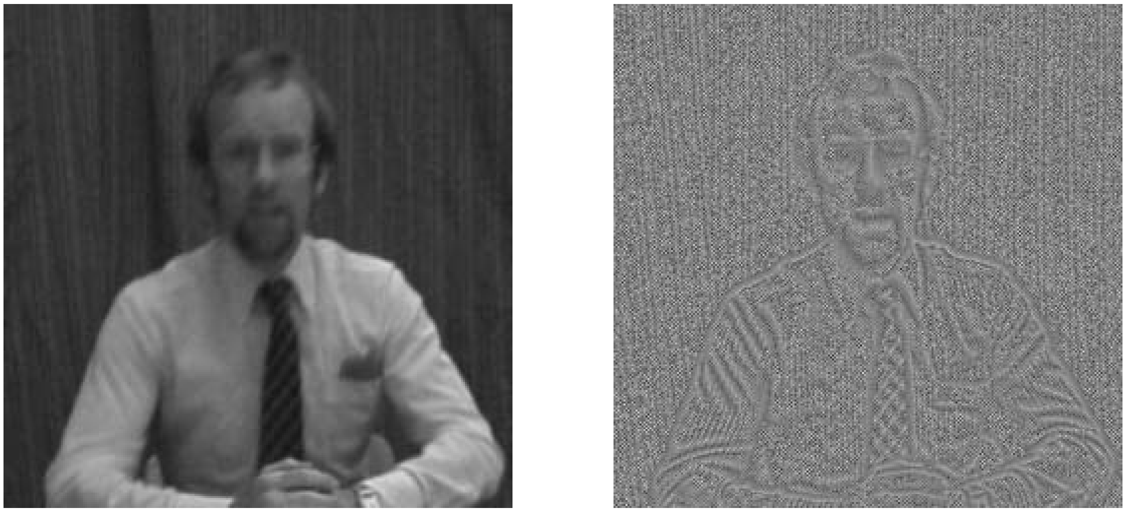


Figure 4.7: Local mean and stretched normalized signal of the “Trevor” sequence with 10dB noise without motion compensation

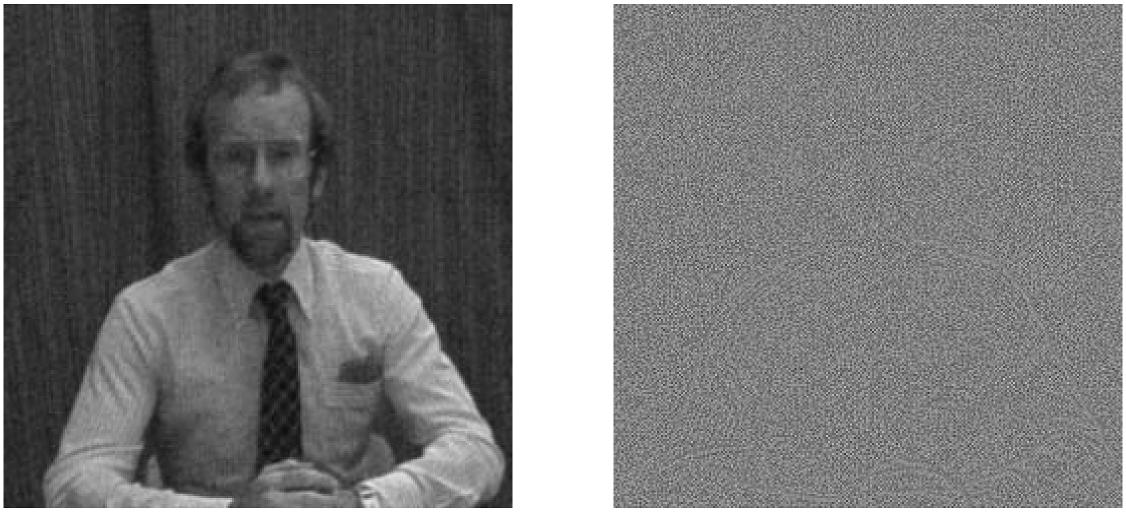


Figure 4.8: Local mean and stretched normalized signal of the “Trevor” sequence with 10dB noise with motion compensation

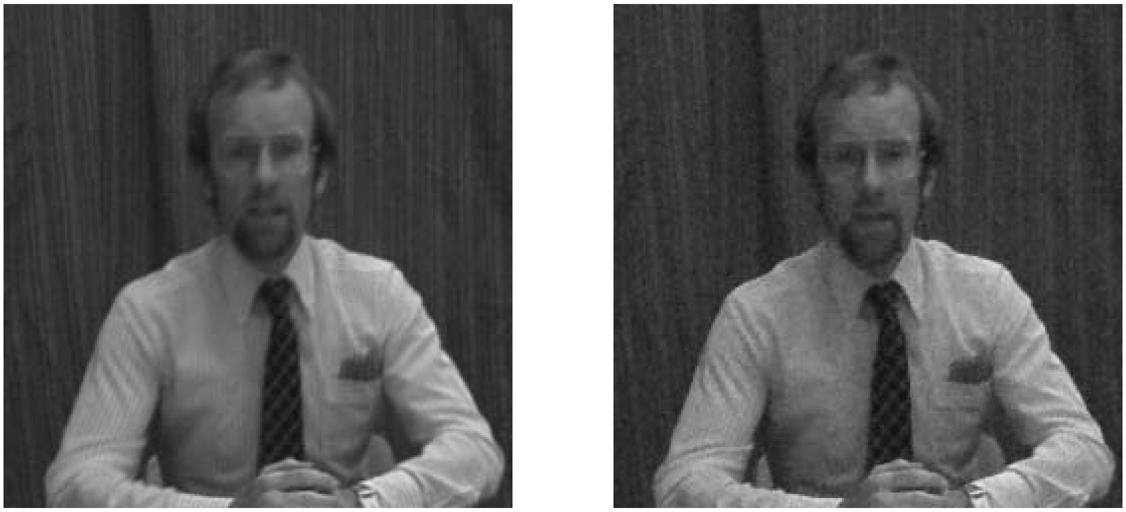


FIGURE 4.9: FILTERED FRAMES WITH THE PROPOSED DECOMPOSITION METHOD WITHOUT (LEFT) AND WITH (RIGHT) MOTION COMPENSATION FOR AN SNR OF 10dB

TABLE 4.2: AVERAGE *improvement in* SNR FOR LAPLACIAN AND GAUSSIAN NOISE AT SNR=10dB

Filter method	*Average Improvement in* SNR (range)	
	Laplacian	Gaussian
Arce	4.8 (0.2)	3.5 (0.2)
Sezan	1.4 (0.3)	4.2 (0.3)
MC+Decomposition	6.9 (0.2)	6.5 (0.4)

Gaussian noise is shown in Figure 4.11. An observed and filtered image of the sequence corrupted with Laplacian noise is shown in Figure 4.12.

For comparison purposes we have processed both corrupted sequences with two methods for general purpose image sequence noise filtering, the spatio-temporal median filter (2.10) by Arce [36], denoted by "Arce" and the LLMMSE filter (2.12),(2.22),(2.23) by Sezan *et al.* [46], denoted by "Sezan". Both methods are reviewed in Chapters 2 and 3. Motion compensation was used with "Sezan" as originally specified. The results are given in Table 4.2. It can be seen that because the proposed method is able to adapt to the noise PDF, it gives good results for both noise sources, while the other methods are sub-optimal with the Laplacian noise. The LLMMSE filter "Sezan" uses box-averaging to establish its parameters. Averaging is only optimal for Gaussian noise and has a large estimation variance for Laplacian noise. The median estimator as used in "Arce" is closer to the optimal estimator for Laplacian noise. However, it has a larger estimation variance in the Gaussian case.

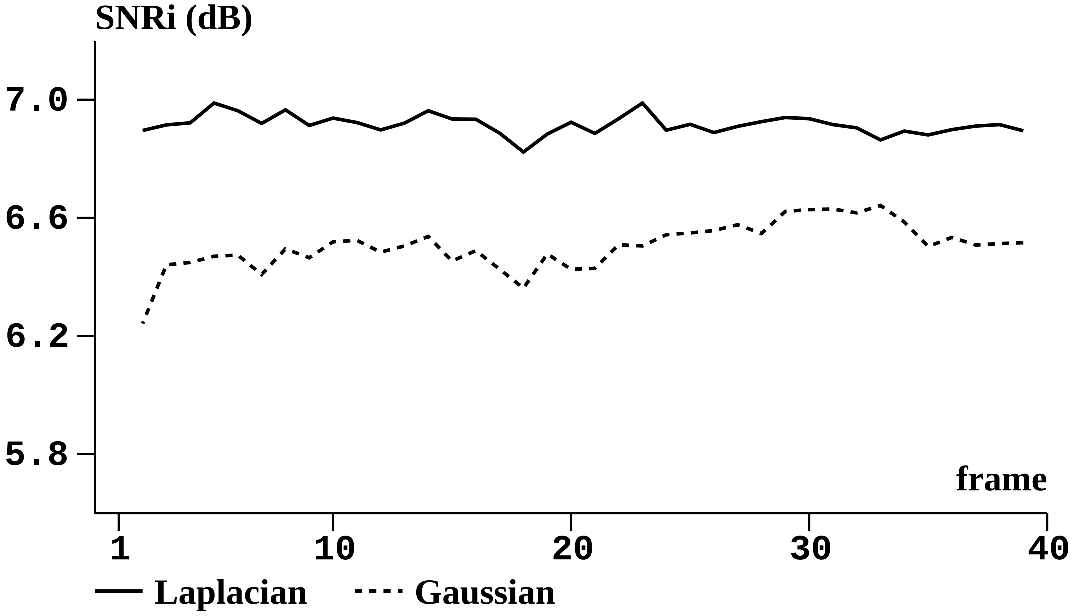


Figure 4.10: Results of the decomposition method with motion compensation for Gaussian and Laplacian noise at 10dB SNR for the “mobile” sequence

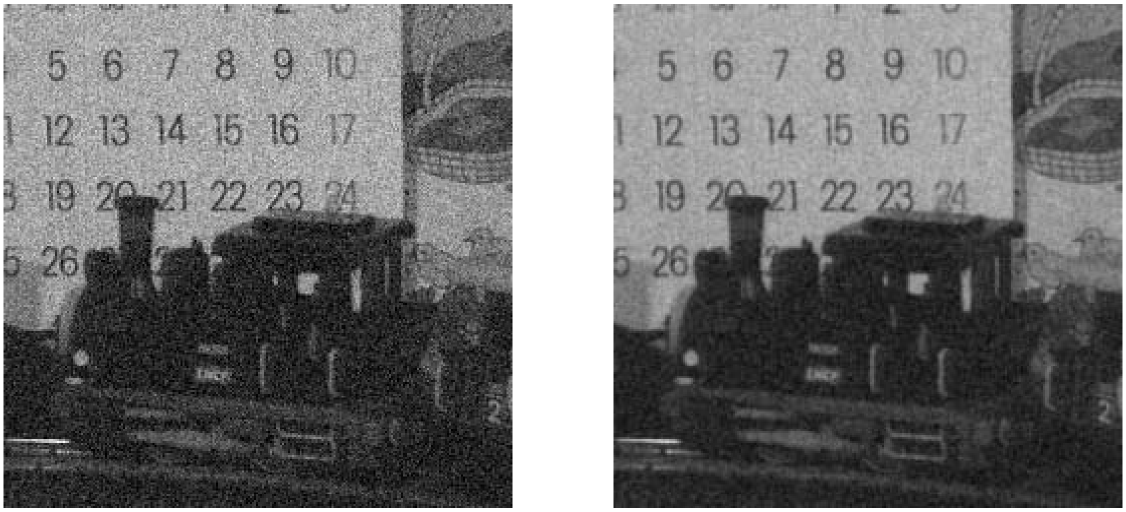


Figure 4.11: Observed image of “mobile” at 10dB Gaussian noise and resulting image of the decomposition method with motion compensation

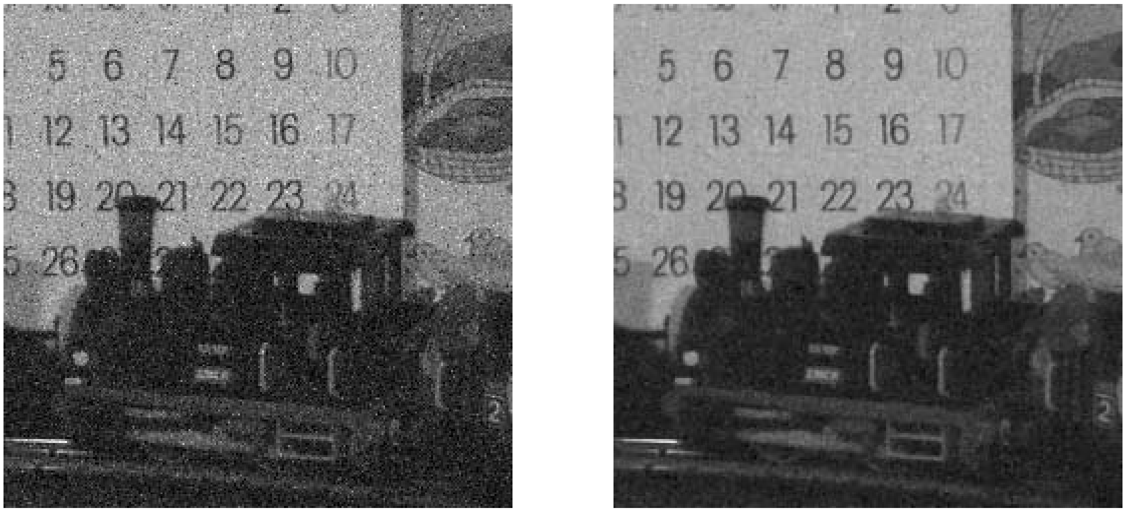


Figure 4.12: Observed image of "mobile" at 10dB Laplacian noise and resulting image of the decomposition method with motion compensation

4.5.3 Discussion

In this chapter we have described an image sequence noise-reduction method that relies on signal normalization prior to filtering. The parameters for this normalization, the local mean and deviation, are estimated by an estimator based on order statistics. This estimator adapts to the current data PDF by an updating process.

We have shown experimentally that the method performs well for several noise levels, both with Gaussian and Laplacian disturbances. We have also seen that a motion-compensating preprocessing step improves the final result which means that the normalization step is not able to remove all the non-stationarities caused by motion. This is likely due to the complex update process which is rather slowly reacting to sudden temporal changes.

Chapter 5

OS-supported noise filtering

In Chapter 4 we have discussed filtering of non-stationary signals from image sequences using trend-removal and normalization. Although the results obtained are promising, the approach has two disadvantages. The first disadvantage is the huge computational effort required for adapting the OS estimators and the noise filter. The second disadvantage is the slow adaptation to changes in the statistics of the spatio-temporal signal. In this chapter, we will improve the usefulness of the algorithm. First, we will reduce the computational effort by removing the residual filter. Second, instead of a recursive update process that tracks the statistics, the OS estimators are now adapted "on the spot".

The omission of the residual filter implies that the final result will be given by a direct combination of the estimates of the local mean and deviation. This requires fast adaptation and high estimation accuracy. These demands are met by adapting the OS estimators to the local signal properties. Only those pixels which are "not significantly different" from the current pixel should be used in the estimation process. The other pixels within the window most likely come from a different object and have no relation to the current pixel.

The qualification of pixels requires detection of outlying observations. Two methods are known to realize this [97]. One is *robust estimation* i.e. a strategy which takes outlying data in the estimation window directly into account in the estimation process and performs a simultaneous detection, ignoration and estimation process. Usually, these methods are more complex than non-robust estimators and involve an iteration process to detect and ignore "outliers" [98]. A second method to acquire accurate estimates from data with outliers is to use a two-step procedure that separates detection and estimation. Prior to estimation, outlying pixels are detected and removed from the data. The estimation is then performed using the remaining reliable data. With this method, conventional estimators can be used. In this chapter both methods will be investigated in the context of image sequence filtering.

The outline of this chapter is as follows. First, in Section 5.1 we simplify the structure of the filter from Chapter 4 and arrive at the well-known LLMMSE filter. As shown in Equation (2.23), this filter needs estimates of the local mean and variance. We then discuss two robust estimation methods for these statistics, namely a simultaneous method consisting of iterative weighed regression in Section 5.2, and a method based on a range test of the data in Section 5.3. The noise-suppression performance of the LLMMSE filter in combination with both robust estimators will be evaluated experimentally in Section 5.4.

5.1 Simplifying the filter structure

The overall filter using the decomposition method from Chapter 4 has the following structure (compare (4.10)):

$$\hat{f}(i,j,k) = \mu_g(i,j,k) + \sigma_g(i,j,k)\ F\left\{\frac{g(i,j,k) - \mu_g(i,j,k)}{\sigma_g(i,j,k)}\right\}. \tag{5.1}$$

Here $F\{\}$ denotes the filtering of the residual signal, where we used a recurrently updated adaptive FIR filter. (The parameters $\mu_g(i,j,k)$ and $\sigma_g(i,j,k)$ are the local mean and local deviance of the observation respectively.)

The method can be simplified and made more computationally efficient by dropping the filter operation and using a simple scalar multiplication $C(i,j,k)$ instead:

$$\hat{f}(i,j,k) = \mu_g(i,j,k) + C(i,j,k)\{g(i,j,k) - \mu_g(i,j,k)\}. \tag{5.2}$$

In the light of the Kalman filters, this structure can be seen as the linear combination of an initial estimate formed by $\mu_g(i,j,k)$ and an update consisting of a fraction of the initial estimation error $g(i,j,k) - \mu_g(i,j,k)$. This Kalman structure is a member of the "switching filters" class as introduced in Chapter 2. The parameter $C(i,j,k)$ controls the filter action from purely averaging if $C(i,j,k) = 0$ to the identity operation, i.e. forwarding the observation, if $C(i,j,k) = 1$.

$C(i,j,k)$ can attain a value which leads to the minimum mean square error in filtering:

$$C(i,j,k)\ \leftarrow\ \min_{C'(i,j,k)}\ E\left\{(f(i,j,k) - \hat{f}(i,j,k))^2\right\}. \tag{5.3}$$

The optimal value for $C(i,j,k)$ is found by setting its derivative with respect to $C'(i,j,k)$ to zero:

$$\frac{dE\{(f(i,j,k) - \hat{f}(i,j,k))^2\}}{dC'(i,j,k)} =$$
$$\sigma_g^2(i,j,k)C'(i,j,k) - \sigma_f^2(i,j,k) - \mu_f^2(i,j,k) + \mu_f(i,j,k)\mu_g(i,j,k) = 0.$$

Here, the property $E\{f(i,j,k)n(i,j,k)\} = 0$ was used. In this expression $\mu_f(i,j,k)$ is the mean of $f(i,j,k)$ and $\sigma_f^2(i,j,k)$ is the variance of $f(i,j,k)$. Because the noise is zero-mean, $\mu_g(i,j,k) = \mu_f(i,j,k)$, and the optimal value for $C'(i,j,k)$ can be written as:

$$C(i,j,k) = \frac{\sigma_f^2(i,j,k)}{\sigma_g^2(i,j,k)}. \tag{5.4}$$

Using $\sigma_g^2(i,j,k) = \sigma_f^2(i,j,k) + \sigma_n^2$, we arrive at:

$$\hat{f}(i,j,k) = \mu_g(i,j,k) + \frac{\sigma_g^2(i,j,k) - \sigma_n^2}{\sigma_g^2(i,j,k)}(g(i,j,k) - \mu_g(i,j,k)). \tag{5.5}$$

This solution can be interpreted as follows: if $f(i,j,k)$ is a constant signal, the variance of $g(i,j,k)$ is equal to σ_n^2 and $C(i,j,k)$ is equal to 0 which causes the filter to become the average operator, the optimal estimator for a constant signal immersed in noise [20, 99]. If no noise is present, $\sigma_n^2 = 0$, and the filter becomes the identity operator.

The derived filter is called the Linear Local Minimum Mean-Square Error (LLMMSE) filter because $C(i,j,k)$ is locally controlled in a minimum mean square-error sense. The LLMMSE filter has been successfully applied to noise filtering of non-stationary images [10, 100, 101] and in image sequence noise filtering [46, 48, 54, 102, 103].

The assumption that an image sequence is locally ergodic makes it feasible to replace the ensemble statistics by the local spatio-temporal statistics. In practical circumstances, $\mu_g(i,j,k)$ and $\sigma_g^2(i,j,k)$ are therefore estimated from a spatio-temporal window surrounding the current pixel (i,j,k). The filtering results that are obtained with (5.5) are however extremely sensitive to the accuracy of the estimates of $\mu_g(i,j,k), \sigma_g^2(i,j,k)$ and σ_n^2 [48, 101].

Problems emerge when choosing an appropriate estimator support because two conflicting requirements have to be satisfied. The first requirement is that the data within the spatio-temporal support has to be fairly homogeneous. This means that the ensemble statistics of pixels contained in this window closely resemble those of $g(i,j,k)$. This usually means a limited support which conflicts with the second requirement that the support of the window has to be large enough to guarantee accurate estimates.

In many applications "box averages" are used, where the statistics are calculated by averaging as [46, 102, 103]:

$$\hat{\mu}_g(i,j,k) = \frac{\sum_{p,q,l \in S} g(i-p, j-q, k-l)}{m}, \tag{5.6}$$

and

$$\hat{\sigma}_g^2(i,j,k) = \frac{\sum_{p,q,l \in S}(g(i-p, j-q, k-l) - \hat{\mu}_g(i,j,k))^2}{m-1}. \tag{5.7}$$

S is the spatio-temporal support of the window of size m. Box averages give inaccurate results if the window contains non-homogeneous data (spatio-temporal edges). The estimate for $\mu_g(i,j,k)$ is then inaccurate and the estimate for $\sigma_g^2(i,j,k)$ is too high. This causes the control parameter $C(i,j,k)$ to approach unity and switch off the filtering action. Switching off filtering in the presence of spatio-temporal edges causes annoying noisy contours of objects in image sequences.

In some applications of the LLMMSE filter, estimators that use a selected homogeneous piece of the spatio-temporal window are applied. Usually, the selection process is aided by the results of an edge detector [10, 101, 104]. Then, if an edge was detected within the window, only the near side of the edge is retained for estimation. The success of this method depends on the robustness and accuracy of the edge detector. It has to be noted that detecting edges is not a trivial task in noisy image sequences [105].

In this chapter, two methods for arriving at accurate estimates are considered. They are both based on observation models which take the outlying observations directly into account without attaching them to image structures such as edges.

5.2 Simultaneous discrimination and estimation: IWLS

The estimators used in Chapter 4 are based on a linear regression through ordered observations. The influence of outliers in regression can be reduced by the use of "robust" regression techniques [97]. In robust regression, outlying observations are detected because they will not fit the model which is imposed on the data. After detection, they can be discriminated in the estimation procedure.

It might be clear that the detection of the outliers is the main problem in robust regression. They can only be identified if the optimal fit is known, which already requires the desired knowledge about the actual position of the outliers. We approach this problem by the selection of the main outliers based on an initial robust regression. These outliers are then assigned weights to reduce their effect on the final result. By iterating this process the weights are adjusted as more outliers are identified. The converged weights are then used to produce the final estimate of $\mu_g(i,j,k)$ and $\sigma_g^2(i,j,k)$. Because of its iterative nature, this procedure is called Iterated Weighed Least Squares (IWLS).

5.2.1 Estimation by regression with ordered observations

The OS estimators for local statistics derived in the previous chapter were based on the assumption that within the acquired sample of size m the local statistics of $\boldsymbol{g}$ do not change. Then, we found the following set of linear relations for the ranks $g_{(r)}(i,j,k)$ of the window centered at (i,j,k):

$$g_{(r)}(i,j,k) = \mu_g(i,j,k) + \sigma_g(i,j,k)p_{(r)} + \epsilon_r(i,j,k), \quad 1 \leq r \leq m, \tag{5.8}$$

where $p_{(r)}$ are the (fixed) rank averages of a parent signal $\boldsymbol{y}$ which is distributed as $q(0,1)$ and $\epsilon_r(i,j,k)$ are the residuals. The statistics $\mu_g(i,j,k)$ and $\sigma_g(i,j,k)$ were found by solving:

$$\hat{\mu}_g(i,j,k), \hat{\sigma}_g(i,j,k) \;\leftarrow\; \min_{\mu_g(i,j,k),\sigma_g(i,j,k)} \sum_{r=1}^{m} \epsilon_r^2(i,j,k). \tag{5.9}$$

It resulted in the following minimum-variance OS estimators for $\boldsymbol{\theta}(i,j,k) = [\mu_g(i,j,k)\ \ \sigma_g(i,j,k)]^T$ (c.f. Eq. (4.23)):

$$\hat{\boldsymbol{\theta}}(i,j,k) = \left[\boldsymbol{A}^T \boldsymbol{C}_{(yy)}^{-1} \boldsymbol{A}\right]^{-1} \boldsymbol{A}^T \boldsymbol{C}_{(yy)}^{-1} \begin{bmatrix} g_{(1)}(i,j,k) \\ \vdots \\ g_{(m)}(i,j,k) \end{bmatrix}, \tag{5.10}$$

where $\boldsymbol{C}_{(yy)}^{-1}$ is the inverse variance matrix of ordered samples from $\boldsymbol{y}$, and $\boldsymbol{A}$ is the design matrix which is composed as follows:

$$\boldsymbol{A} = \begin{bmatrix} 1 & p_{(1)} \\ \vdots & \vdots \\ 1 & p_{(m)} \end{bmatrix}. \tag{5.11}$$

5.2.2 Selecting and rejecting outlying observations

So far, we have assumed that within the data set pixels from only one parametrized PDF with the same parameters are present. If pixels with other statistics are present within the data set, the estimators must have a selection method to be able to use only information from the same parametrized PDF.

The difference in the statistics of the outlying pixel can be used for detection. Given a robust fit, i.e. a fit through points that are related to certified pixels will cause the outliers to have relatively high residual errors (see Figure 5.1). These residual errors can be used to detect outliers.

Large residual errors have a substantial influence on the criterion (5.9) used for derivation of the estimator (5.10). The aim in robustifying the estimator is to

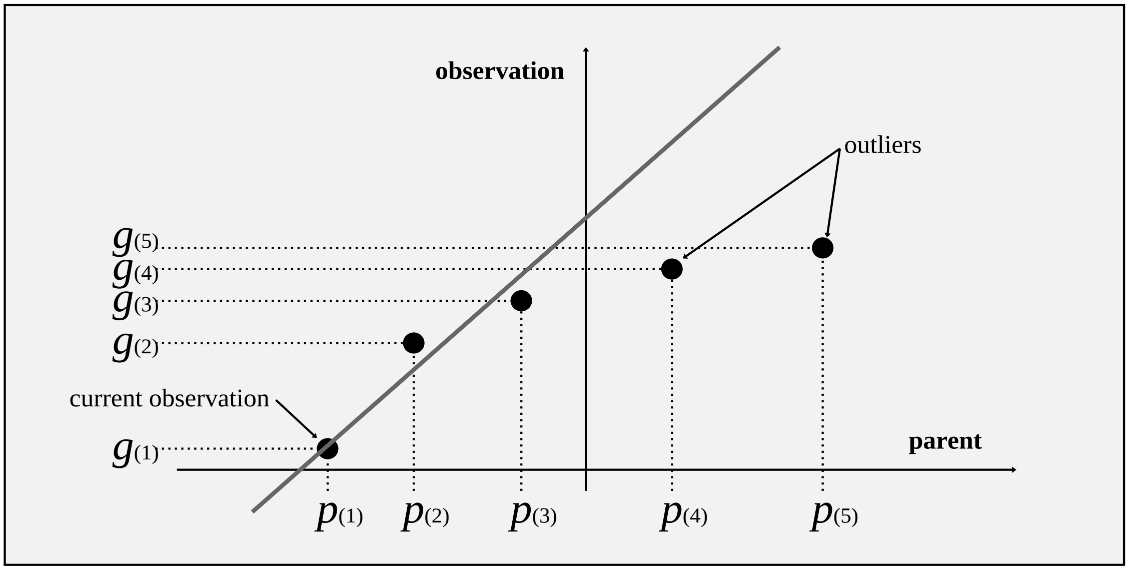


Figure 5.1: The large residual errors that some observed pixels give in a robust fit, arise because these pixels have other ensemble statistics than the current one that governs the robust fit.

reduce the influence of outliers on the criterion function and to compose a method to detect them based on the same criterion. These requirements are conflicting for most criterion functions. A criterion function that can fulfill these demands is the following non-linear function [97, 98, 106]:

$$\hat{\mu}_g(i,j,k), \hat{\sigma}_g(i,j,k) \;\leftarrow\; \min_{\mu_g(i,j,k),\sigma_g(i,j,k)} \; \sum_{r=1}^{m} \Psi\{\epsilon_r(i,j,k)/\sigma\}. \tag{5.12}$$

Where the residual error $\epsilon_r(i,j,k)$ was:

$$\epsilon_r(i,j,k) = g_{(r)}(i,j,k) - \begin{bmatrix} 1 & p_{(r)} \end{bmatrix}^T \hat{\boldsymbol{\theta}}(i,j,k). \tag{5.13}$$

The deviation of the residuals is given by a robust measure σ. Division by this measure is necessary to make the non-linear function $\Psi\{\}$ scale insensitive.

The resulting robust estimates are found by differentiating Equation (5.12) with respect to $\hat{\boldsymbol{\theta}}(i,j,k)$ and forcing the result to zero:

$$\sum_{r=1}^{m} \begin{bmatrix} 1 & p_{(r)} \end{bmatrix}^T \psi\{\epsilon_r(i,j,k)/\sigma\} \;=\; \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \tag{5.14}$$

where $\psi\{\}$ is the derivative of $\Psi\{\}$. By introducing the following weights:

$$w_r\{\hat{\boldsymbol{\theta}}(i,j,k)\} = \frac{\psi\{\epsilon_r(i,j,k)/\sigma\}}{\epsilon_r(i,j,k)/\sigma}, \tag{5.15}$$

equation (5.14) can be rewritten as:

$$\sum_{r=1}^{m} w_r\{\hat{\boldsymbol{\theta}}(i,j,k)\} \begin{bmatrix} 1 & p_{(r)} \end{bmatrix}^T \epsilon_r(i,j,k) \;=\; \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{5.16}$$

The scalar $w_r\{\hat{\boldsymbol{\theta}}(i,j,k)\}$ can be seen as a weight indicating the outlying nature of observation $g_{(r)}(i,j,k)$ based on a given estimate.

Equation (5.16) is non-linear through $\psi\{\}$ and is therefore usually solved by an iterative procedure. We propose to use IWLS, where the following variant of (5.16):

$$\sum_{r=1}^{m} w_r\{\hat{\boldsymbol{\theta}}^s(i,j,k)\} \begin{bmatrix} 1 & p_{(r)} \end{bmatrix}^T \epsilon_r(i,j,k) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \tag{5.17}$$

is solved for $\hat{\boldsymbol{\theta}}^{s+1}$. The iteration step is denoted by s and $\epsilon_r(i,j,k)$ is the residual of the next estimate $\hat{\boldsymbol{\theta}}^{s+1}$:

$$\epsilon_r(i,j,k) = g_{(r)}(i,j,k) - \begin{bmatrix} 1 & p_{(r)} \end{bmatrix}^T \hat{\boldsymbol{\theta}}^{s+1}(i,j,k). \tag{5.18}$$

Performing linear regression on (5.17) results in the following IWLS procedure [98, 106]:

$$\hat{\boldsymbol{\theta}}^{s+1}(i,j,k) = \left[\boldsymbol{A}^T\boldsymbol{C}_{(yy)}^{-1}\boldsymbol{W}\boldsymbol{A}\right]^{-1}\boldsymbol{A}^T\boldsymbol{C}_{(yy)}^{-1}\boldsymbol{W}\begin{bmatrix} g_{(1)}(i,j,k) \\ \vdots \\ g_{(m)}(i,j,k) \end{bmatrix}. \tag{5.19}$$

Note that the differences with the non-robust estimator from Equation (5.10) lie in the diagonal weighing matrix $\boldsymbol{W}$:

$$\boldsymbol{W} = \text{diag.}\left[w_r\{\hat{\boldsymbol{\theta}}^s(i,j,k)\}\right] \tag{5.20}$$

The robust algorithm usually requires a few iterations to converge to a robust estimate $\hat{\boldsymbol{\theta}}(i,j,k)$. A suitable robust initial estimate for $\hat{\boldsymbol{\theta}}(i,j,k)$ is:

$$\hat{\boldsymbol{\theta}}^0(i,j,k) = \begin{bmatrix} g(i,j,k) & \sigma_n \end{bmatrix}^T. \tag{5.21}$$

This means that the initial estimate for $\mu_g(i,j,k)$ is the current observation. The known noise deviation is used as an initial deviation estimate. This is based on the assumption that the deviance of the original signal is fairly small compared to the noise deviance within an object which will have a relatively fixed intensity. An example of an initial fit for $m = 5$ is shown in Figure 5.1, where the current observation, in this case the smallest, acts as a leverage point.

Important aspects in the robust estimator are the estimate of σ and the function $\Psi\{\}$. The residual deviation σ is estimated from $\boldsymbol{\epsilon}(i,j,k)$ by:

$$\hat{\sigma} = \frac{\text{median}\{|\epsilon_1|, |\epsilon_2|, \cdots, |\epsilon_m|\}}{0.6745}, \tag{5.22}$$

which was shown by Hampel [107] to be the most robust estimate of deviation. This estimate is repeated after each iteration based on the new residual errors $\epsilon_r(i,j,k)$.

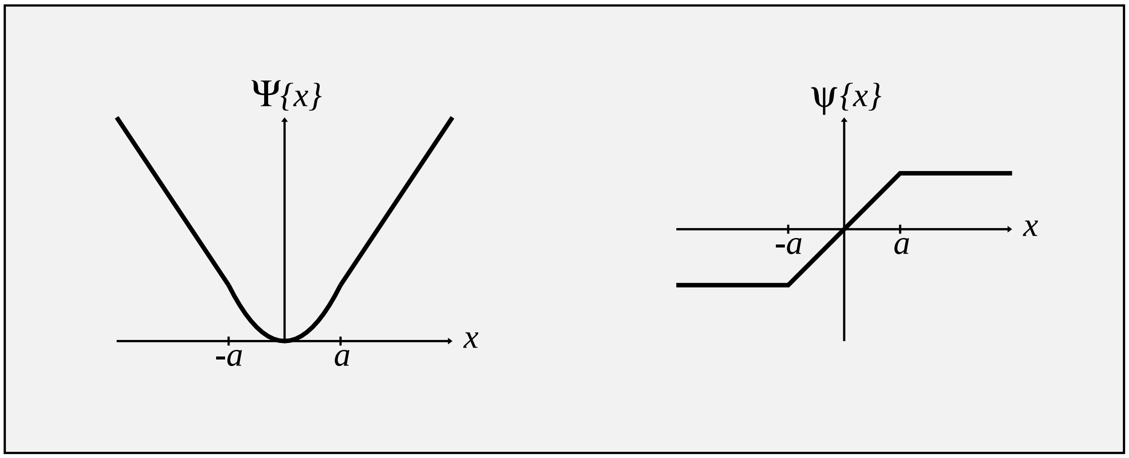


Figure 5.2: The Huber criterion function and its derivative

The function $\Psi\{\}$ has to fulfill some requirements, it has to be symmetric: $\Psi\{x\} = \Psi\{-x\}$ and it has to have a unique minimum at $\Psi\{0\}$. Further, it has to be convex to guarantee convergence of (5.19) [97, 108]. The following $\Psi\{\}$ and $\psi\{\}$, which were derived by Huber [98, 109], will lead to robust estimates (see also Figure 5.2):

$$\Psi\{x\} = \begin{cases} \frac{x^2}{2}, & |x| \leq a \\ a|x| - \frac{a^2}{2}, & |x| > a \end{cases} \quad ; \quad \psi(x) = \begin{cases} -a, & x < -a \\ x, & |x| \leq a \\ a, & x > a \end{cases} . \tag{5.23}$$

This specification for $\Psi\{\}$ is associated with a Gaussian shape in the middle and with "double exponential" tails. With Huber's function $\Psi\{\}$, $\psi\{\}$ and $\hat{\sigma}$, the tuning constant a would be given a value of about 1.5. The reason for this selection is that if $\boldsymbol{\epsilon}(i,j,k)$ actually comes from a Gaussian distribution, most of the residual errors would obey the property that $|\epsilon_r(i,j,k)| \leq 1.5\sigma$. If all residual errors satisfy this inequality, then for all ranks: $\psi\{\epsilon_r(i,j,k)/\sigma\} = \epsilon_r(i,j,k)/\sigma$ and the regular least squares solution will arise [98].

The derivation of the asymptotic efficiency of the resultant estimator is rather complex because of the non-linearity of $\Psi\{\}$ and the difficulty to establish the degrees of freedom in $\hat{\sigma}$. It has been established in [98, 110] that if σ is known, the choice $a = 1.345$ is needed to achieve an asymptotic efficiency of 95% under Gaussian assumptions. When the uncertainty in the value of $\epsilon_r(i,j,k)/\sigma$ rises if σ has to be estimated from a limited number of residual errors in a robust way, then using $a = 1.5$ will result in a comparable asymptotic efficiency [98]. An overview of the IWLS procedure embedded within the LLMMSE filter is given in Figure 5.3.

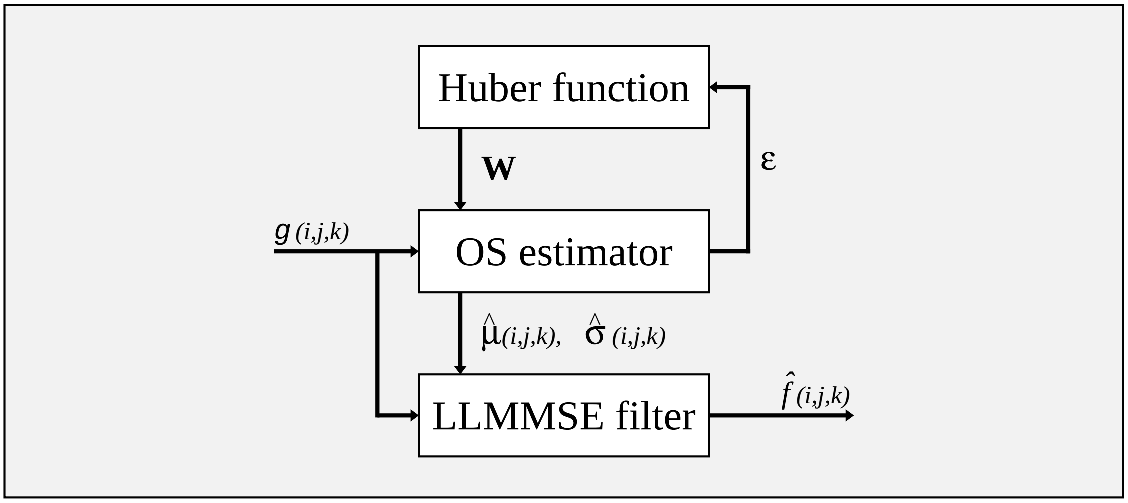


Figure 5.3: An overview of the LLMMSE filter supported by robust IWLS estimators based on order statistics.

5.3 Rejection prior to estimation: the Range Test

This section investigates another method to remove outlying observations, namely detection and rejection prior to the estimation of $\mu_g(i,j,k)$ and $\sigma_g^2(i,j,k)$. The selection is done by performing tests of similarity on the pixel values within the data set. Only those pixels which pass these tests are used for subsequent estimation. First, we discuss the use of regular tests such as the t-test to select pixels that are *not significantly different*. We will show that the usefulness and the validity of these tests is limited in our application. Therefore, we will subsequently investigate the use of the Studentized range as an efficient and effective measure of similarity between groups of pixel values.

5.3.1 Testing pixel values for similarity

Selecting outlying observations prior to estimation is performed by comparing the intensity of the current pixel with that of candidate pixels included in the sample. A strict test would be equality, which selects only pixels with the same intensity, but obviously, this will not be useful for estimation. As the noise has influenced the observed intensity, the strict equality demand has to be relaxed, which means accepting that pixels with "comparable" intensities are likely to belong to the current object.

A mechanism has to be devised where perturbation by noise only will not result in rejection of the tested pixel. The design of a test is based on the observation model and its connected assumptions. Tests will be formulated as hypothesis tests in this section. For instance, the hypothesis that pixel a is similar to pixel b, will be subject to testing and be accordingly accepted or rejected.

We assume the additive, uncorrelated and zero-mean noise model with $q\{\}$ as the PDF of the noise. An additional assumption for testing is that objects which occur in the original sequence have a constant gray level within the spatio-temporal window. Both assumptions suggest that, within the window, the observations $g(i,j,k)$ belonging to the same object are independently distributed with the same mean $f(i,j,k)$:

$$g(i,j,k) \sim q\{f(i,j,k), \sigma_n\}. \tag{5.24}$$

Let us denote the pixel values from the spatio-temporal window as: $g_1, g_2, \cdots, g_m$ (without any specific order) with g_c as the current pixel value of $g(i,j,k)$. Each pixel g_l is then also distributed according to (5.24):

$$g_l \sim q\{f_l, \sigma_n\}, \;\; 1 \leq l \leq m. \tag{5.25}$$

A prior test can be performed to investigate the hypothesis that the window contains no outlying pixel values. Based on (5.24), this test involves the following hypothesis:

$$H: \;\; f_1 = f_2 = \cdots = f_c = \cdots = f_{m-1} = f_m. \tag{5.26}$$

The test can be performed by an F-*ratio* test [111]. Note that testing H will only show whether *all* pixels are not significantly different, but outlying pixels are not selected. Therefore, the F-*ratio* test has to be repeated for every combination of the current pixel and the other pixels from the window, which is a huge task, even for moderate m.

A popular test to investigate whether pixels are significantly different or not is the t-test. Two pixels g_c and g_l are declared "significantly different" if:

$$\frac{|g_c - g_l|}{\sigma_n} \; > \; \sqrt{2} t_{\alpha/2}, \tag{5.27}$$

where $t_{\alpha/2}$ is the upper $\frac{1}{2}\alpha$ significance point whose value depends on $q\{\}$. They are readily tabulated for the Gaussian distribution [112]. As the parameter α is the desired "false alarm" probability, the t-test has a reliability level of $100(1-\alpha)\%$.

If all pixels g_l within the sample ($1 \leq l \leq m$) are tested for similarity with the current observation g_c, the probability of making a correct *overall* decision is:

$$P = p\left\{\frac{g_{max} - g_{min}}{\sigma_n} \leq \sqrt{2} t_{\alpha/2}\right\}, \tag{5.28}$$

where $p\{\}$ denotes probability and g_{max} and g_{min} are the extremes of the accepted set of m' pixel values. It appears that this probability P of making a correct overall decision declines with increasing m' [96, 113]:

$$P = (1-\alpha)^{m'-1}. \tag{5.29}$$

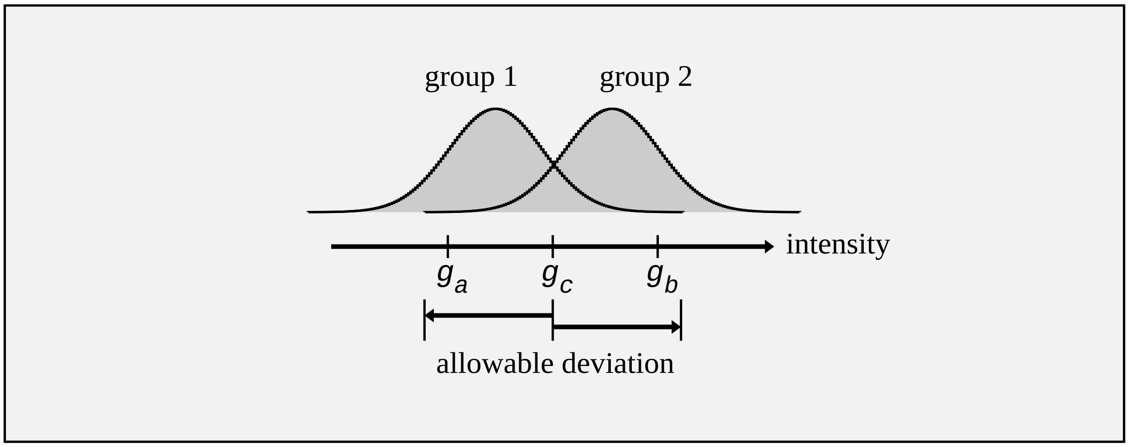


FIGURE 5.4: USING THE t-TEST, INTENSITIES g_a AND g_b ARE FOUND TO BE NOT SIGNIFICANTLY DIFFERENT FROM THE OBSERVATION g_c AND BOTH WILL BE USED IN THE ESTIMATION PROCESS. HOWEVER, THIS ACCEPTANCE DOES NOT MEAN THAT g_a AND g_b ARE REGARDED NOT SIGNIFICANTLY DIFFERENT USING THE SAME TEST. SO THE UNIFORMITY OF A GROUP OF INTENSITIES THAT PASSES THE t-TEST IS QUESTIONABLE.

As a result, the probability $1 - P$ of at least one incorrect decision in the selected set will be much larger than α. This phenomenon is illustrated in Figure 5.4 for a Gaussian noise distribution. Here, both g_a and g_b are not regarded significantly different to g_c in individual tests because both values lie within a given distance from g_c. However, comparing g_a with g_b using the same test results in the decision that g_a and g_b are significantly different. Yet, the above test considers them both reliable enough for estimation purposes.

5.3.2 Using the range in testing

To overcome the problems with the t-test if decisions have to be made for a larger number of pixels, Duncan suggested the "modified range test" [114]. A range is a function of the ordered statistics. If the pixel values from the spatio-temporal window are arranged in increasing values: $g_{(1)} \leq g_{(2)} \leq \cdots \leq g_{(m)}$, then the difference $g_{(m)} - g_{(1)}$ is called the *sample range*. The absolute difference of two arbitrary pixels $|g_{(a)} - g_{(b)}|$ is simply called a range. The ratio of a range to an independent "root-mean-square estimate" of the population standard deviation (here σ_n) is called a *Studentized* range [113].

The Duncan method tests various Studentized ranges within a sample with the probability $(1-\alpha)$ of making an overall correct decision for the selected set. An additional advantage is the relatively low computational burden of Duncan's method.

The specific algorithm, involving the range test to select pixels belonging together proceeds as follows. A pixel value $g_{(a)}$ is declared not significantly different from $g_{(b)}$ if:

$$\frac{|g_{(a)} - g_{(b)}|}{\sigma_n} \leq \rho_{|a-b|,\alpha}. \tag{5.30}$$

Where $\rho_{|a-b|,\alpha}$ is the upper α significance point for a Studentized range over $|a-b|$ values. These significance points are tabulated in literature [96, 113]. Figure 5.5 gives an idea of the values for $\alpha = 5\%$ and a known value for σ_n for a number of set sizes. If a range of pixels passes the test, all pixels covered by this range are automatically regarded to be not significantly different. Of course, the opposite is not true. If the range fails the test, the pixel values covered by the range have to be considered more closely.

The ultimate goal of the Range Test is to select the largest subset of pixels including the current observation that is homogeneous. More formally; if $a \leq b$:

$$\begin{aligned} g_{(a)}, \cdots, g_{(c)}, \cdots, g_{(b)} \leftarrow \max_{a',b'} (b-a) \qquad & \text{subject to: } \frac{|g_{(a)} - g_{(b)}|}{\sigma_n} \leq \rho_{(b-a),\alpha}, \\ & \text{and: } g_{(a)} \leq g_{(c)} \leq g_{(b)} \end{aligned} \tag{5.31}$$

Using the properties of the Range Test and the ordered structure of the data in the entire window means that outliers can be detected in an efficient way. For instance, first the range spanned by $g_{(1)}$ and $g_{(m)}$ is tested for homogeneity. Then, based on the specific results, the test can be stopped or tests can be performed on ranks spanning smaller subsets. The algorithm continues until the first large homogeneous subset, including the current observation is found.

To decrease the table dimension, the calculation effort, and to overcome some inconsistencies if multiple pixels with similar values are tested, pixels with the same value are placed on a single rank, as in the following ordering:

$$g_{(1)}^{m_1}, g_{(2)}^{m_2}, \cdots, \cdots, g_{(m)}^{m_m}, \quad g_{(r)} < g_{(r+1)} \quad \forall\, 1 \leq r < m, \tag{5.32}$$

where the superscripts m_r denote the number of pixels available on rank r. Note also that no equalities will occur anymore. This method of ordering is especially efficient if the data has limited accuracy.

With this new ordering, a slight modification of the test is necessary. The pixels within the range spanned by a and b are then regarded not significantly different if:

$$\frac{|g_{(a)}^{m_a} - g_{(b)}^{m_b}|}{\sigma_n} \leq \sqrt{\frac{m_a + m_b}{2 m_a m_b}} \rho_{|a-b|,\alpha}. \tag{5.33}$$

The Range Test will specify a sub-set of pixel values from the operation window containing the current observation. An example of the operation of the Range Test on non-stationary image data is illustrated in Figure 5.6. In the left image the

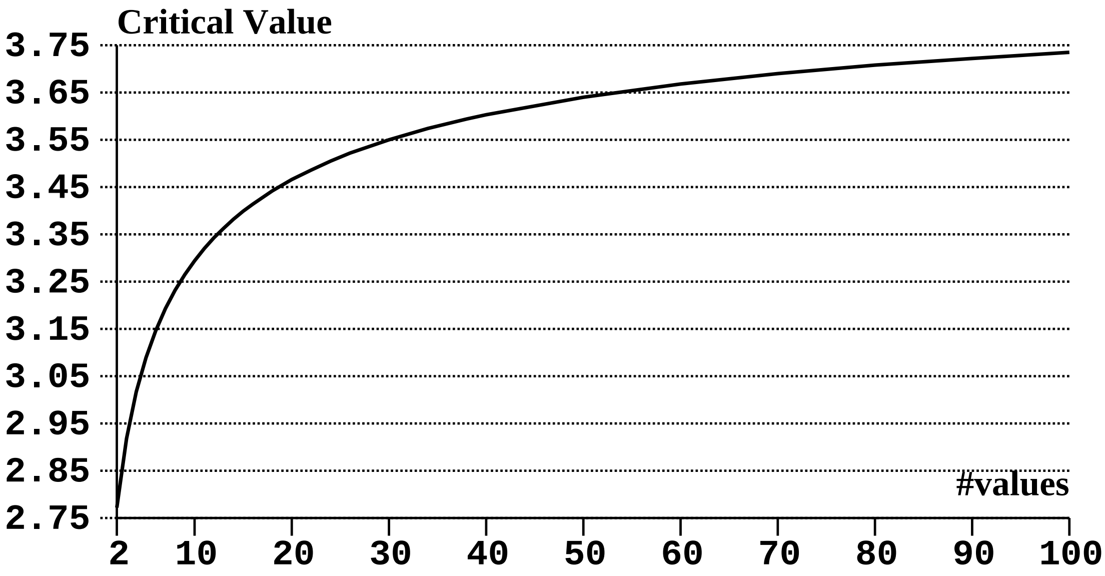


FIGURE 5.5: THE CRITICAL VALUES $\rho_{|a-b|}$ FOR THE RANGE TEST ON GAUSSIAN DATA, FOR A SIGNIFICANCE LEVEL OF $\alpha = 5\%$, VARIOUS NUMBER OF VALUES $|a - b|$, AND A KNOWN VALUE FOR σ_n

square spatial window was placed just before the character. The white sections give the pixels from the background that are selected. Note that even pixels separated by the character are selected. In the right image the spatial window was centered on the upper right part of the character and only pixels from the character are selected. Note that in this example, a spatial window was employed. In image sequences, the selection is performed from a spatio-temporal window.

5.4 Experimental evaluation

An experimental evaluation of both OS supported LLMMSE filters, one supported by the Iterated Weighed Least Squares (IWLS) algorithm and one by the Range Test (RT) is considered in this section. Both methods are evaluated with and without motion compensation. As object sequences we have used frames 1 to 40 of the “mobile” sequence and frames 21 to 50 of the “Trevor” sequence at noise levels of 10 and 20 dB. The noise was white, Gaussian, invariant and independent of the original signal. The results of the IWLS method were already listed in Chapter 2 for the un-compensated and Chapter 3 for the motion-compensated case, specified as “Kleihorst” and “Kleihorst-MC” respectively.

In both methods, the spatio-temporal window used for estimating the local statistics for the LLMMSE filter was a spatio-temporal cube of 27 pixels centered at the

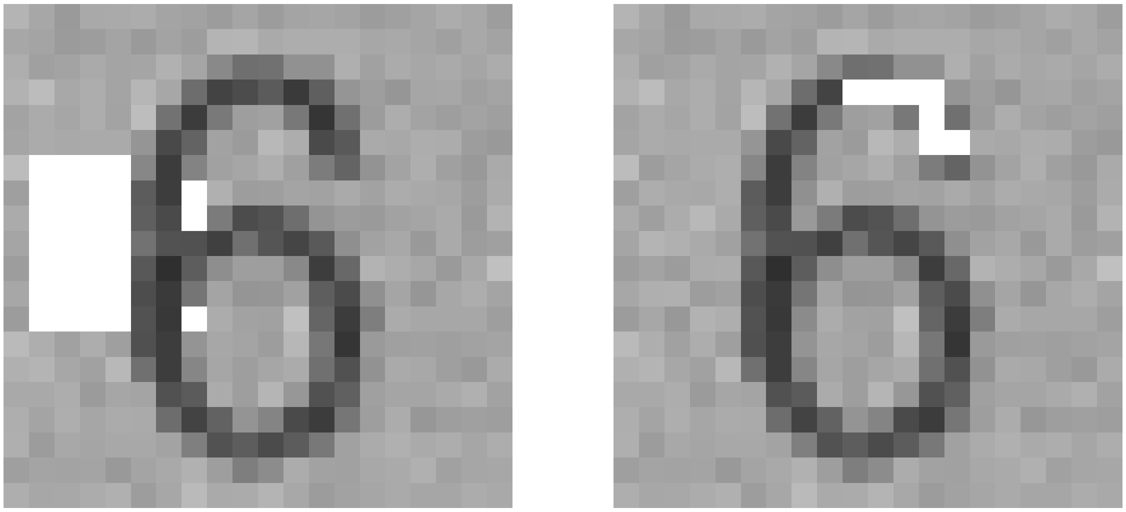


Figure 5.6: A visualization of the selective properties of the Range Test. In the left image, the operating window is centered in the background and pixels from the background (white) are selected for filtering. In the right image, the operating window is selected on a portion of the character. It can be seen that pixels from the character (white) are selected for filtering.

Table 5.1: Average *improvement in* SNR for the "mobile" sequence.

Outlier Selection	*Average Improvement in* SNR (range)	
	SNR=10dB	SNR=20dB
IWLS	7.4 (0.2)	3.9 (0.1)
IWLS-MC	7.6 (0.2)	4.0 (0.1)
RT	7.5 (0.3)	3.6 (0.1)
RT-MC	8.1 (0.2)	3.8 (0.1)

current pixel. In the IWLS method the parent PDF that defines the estimators was fixed at a Gaussian shape. The parameter in Huber's functions for discriminating outliers was set to $a = 1.5$ resulting in an estimator efficiency of around 95%. The iteration process was stopped if the LLMMSE filter output had converged within an integer value, which always occurred within 5 iterations. The Range Test used the decision table presented in Figure 5.5, which is for Gaussian noise, based on a critical value of $\alpha = 5\%$. The known noise deviance was used to create Studentized ranges.

The results are presented in Table 5.1 for the "mobile" sequence and in Table 5.2 for the "Trevor" sequence. In these tables "IWLS" and "IWLS-MC" denote the LLMMSE filter supported by IWLS estimation without and with motion compensation, respectively. "RT" and "RT-MC" denote the LLMMSE filter supported

TABLE 5.2: AVERAGE *improvement in* SNR FOR THE "TREVOR" SEQUENCE.

Outlier selection	*Average Improvement in* SNR (range)	
	SNR=10dB	SNR=20dB
IWLS	7.2 (0.4)	2.8 (0.3)
IWLS-MC	7.3 (0.5)	2.4 (0.5)
RT	7.3 (0.6)	2.1 (0.4)
RT-MC	7.6 (0.6)	2.3 (0.4)

by an estimation with selection by the Range Test without and with motion compensation, respectively. The improvement figures of both selection methods can be compared because both IWLS and RT are tuned by a and α to the same efficiency. From the tables, it appears that the LLMMSE filter gives better result with the RT outlier selection (i.e. prior to estimation) for the low SNR of 10dB. For 20dB, the IWLS outlier selection (i.e. simultaneous selection and estimation) gives better results. This effect is present in both sequences. A probable explanation is the strong model-based nature of the IWLS process that exploits the relations between ordered observations. The RT selection method only uses the observation model and assumes that $f(i, j, k)$ is constant within the window. It can be noticed from the tables that motion compensation will only slightly increase the SNRi. This means that we have established adaptation methods that reduce the need for motion compensation.

The SNRi curves are shown in Figure 5.7 for the IWLS selection and in Figure 5.8 for the RT selection, respectively. From the relatively smooth character of the results without motion compensation it can be seen that the OS supported LLMMSE filters are able to handle the non-stationary signals reasonably without motion compensation. In addition, this observation is supported by the great similarity of the un-compensated and compensated results. Overall, the noise suppressing characteristics of the LLMMSE filter are relatively high and consistent as can be seen from the results on both sequences.

From a visual point of view, the noise is substantially removed without unreasonably affecting the sharpness of the sequences and without blurring moving objects. This can be seen from a result image of the LLMMSE filter supported by the Range Test for the "mobile" sequence in Figure 5.9 where both the observation at 10dB SNR and the filter result are shown. The filter result on the "Trevor" sequence is shown in Figure 5.10.

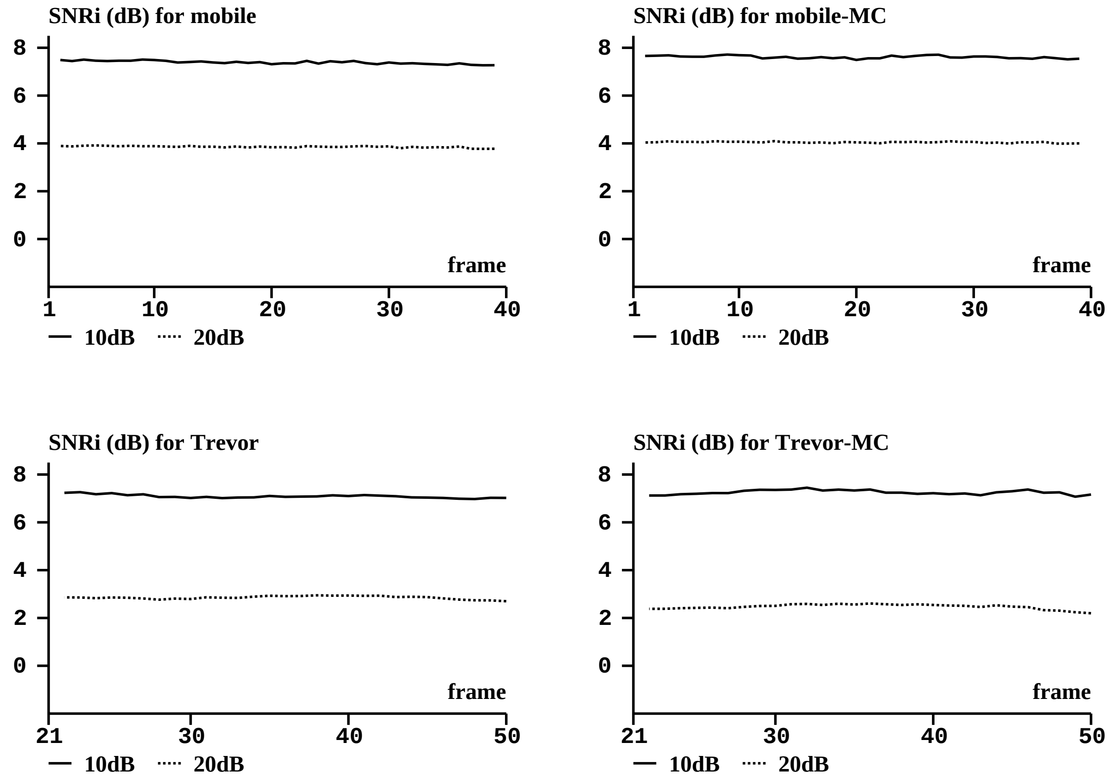


Figure 5.7: Experimental results for the LLMMSE filter with outlier selection by the IWLS method

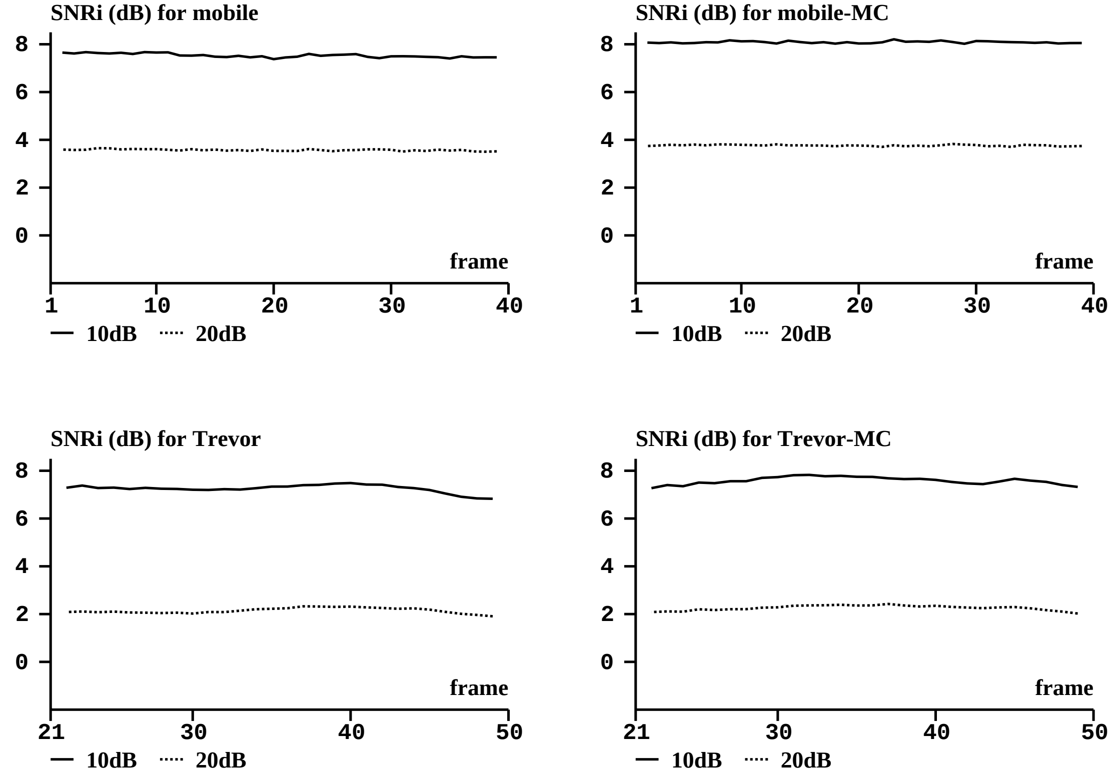


Figure 5.8: Experimental results for the LLMMSE filter with outlier selection by the Range Test

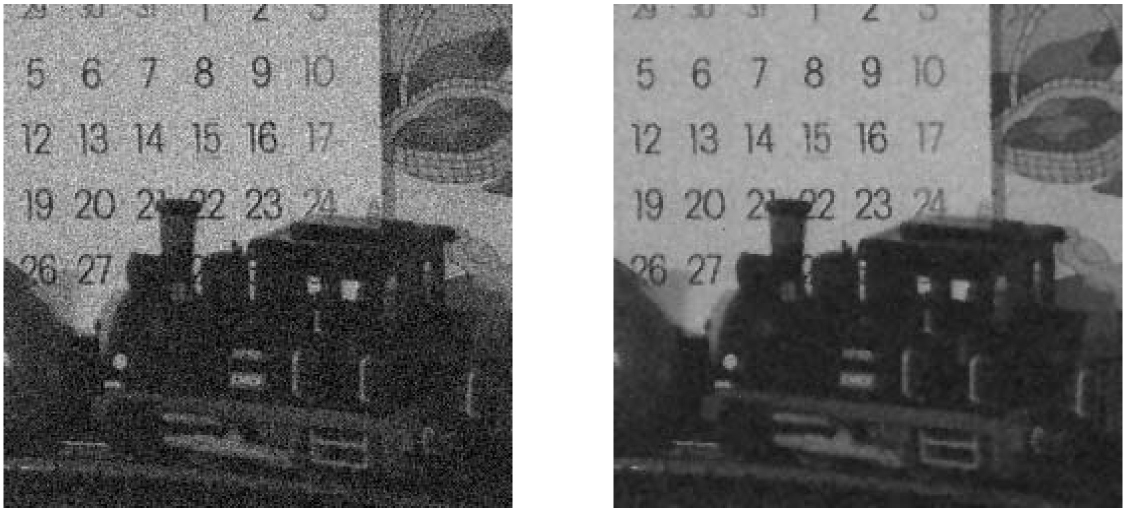


Figure 5.9: The observed "mobile" sequence at 10dB SNR and the result after applying the LLMMSE filter supported by the Range Test

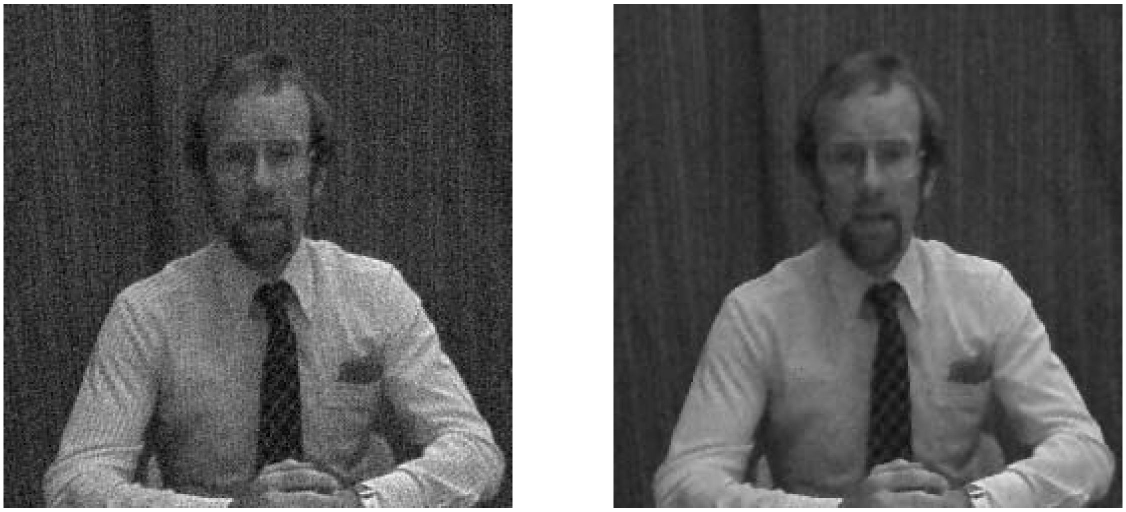


Figure 5.10: The observed "Trevor" sequence at 10dB SNR and the result after applying the LLMMSE filter supported by the Range Test

Chapter 6

Applications to signal-dependent noise

6.1 Introduction

The filters discussed in the previous chapters were designed to operate on observation models with additive, signal-independent noise. Although these filters will most certainly have many practical applications, a number of practical situations result in noise which is *signal dependent.* The signal dependency manifests itself in the statistical expressions of the observation noise.

In this chapter we consider two practical problems of interest in which the observation noise is signal dependent, namely quantum-limited imaging, for instance with low-dosage X-ray imaging, and gamma-corrected video signals. The quantum-limited image-formation process suffers from noise because the observation intensity depends on the stochastic arrival times of photons. The uncertainty (noise) in the number of photons captured in a given period depends on the photon rate which is influenced by the original intensity. Signal-dependent noise is also encountered in gamma-corrected video signals. Here, the non-linear gamma correction introduces a signal-dependent mapping of the sensor noise in the observed signal.

We will derive specific filters, using techniques from the previous chapters for these two practical applications. The main focus is on the implications of signal-dependent noise for the OS estimators. In Section 6.2, we consider the filtering of quantum-limited images as available from clinical X-ray image sequences. In Section 6.3, we consider the filtering of gamma-corrected video signals with application to removing noise caused by the electronics in the camera. In Section 6.4 we perform an experimental evaluation using synthetic and real sequences.

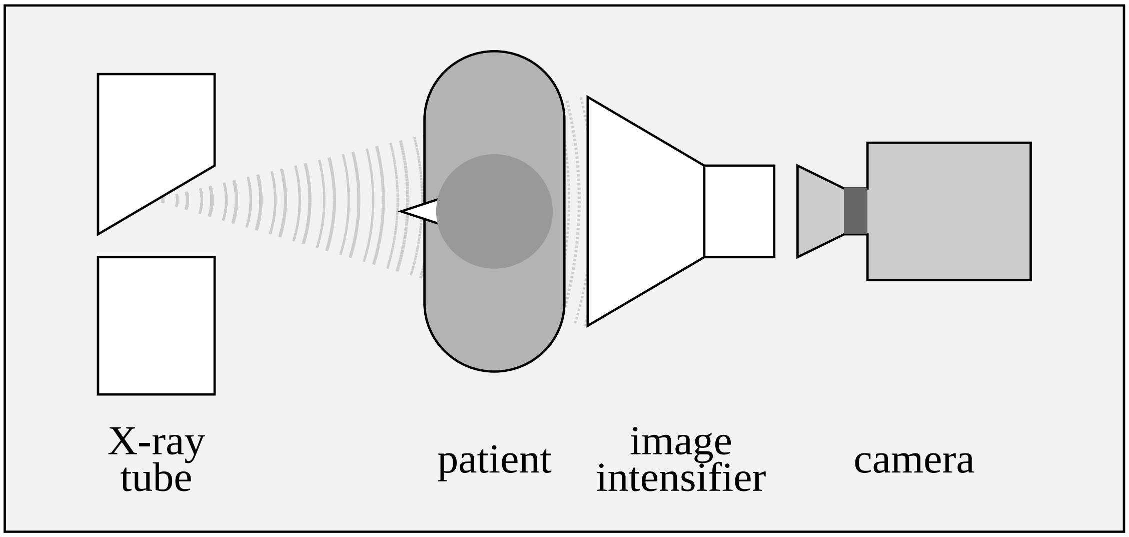


FIGURE 6.1: THE TYPICAL COMPONENTS OF A DIAGNOSTIC X-RAY SYSTEM FOR MEDICAL PURPOSES

6.2 Quantum-limited image sequences

6.2.1 Clinical X-ray imaging

Image intensities are generally the result of photons captured by an image-scanning device. In most cases the number of photons captured in the area spanned by a single pixel is superfluous. However, under some practical circumstances, this number is rather low yielding so-called *quantum-limited* images. The quantum-limited imaging process suffers from the stochastic nature of the photon arrival rate which manifests itself as noise, called "quantum noise" or "quantum mottle" [103, 115]. An intrinsic property of the quantum-limited imaging process is the dependency of the noise characteristics on the signal intensity. This property necessitates the use of special filter techniques.

Quantum-limited images are encountered in a number of practical situations: in astronomical imaging, where the exploration of remote galaxies is realized by capturing the few radiated photons of remote stars; in night-vision systems, where because of darkness, the entire scene radiates a limited amount of photons; also, in clinical X-ray imaging where the number of photons is limited as a result of lowering the radiation exposure. We focus on quantum noise in clinical X-ray sequences.

The characteristic components of an X-ray system are shown in Figure 6.1. An X-ray tube generates a beam of X-rays. Upon passing the patient, the intensities of these rays are reduced depending on tissue properties, in this way creating a varying/modulated X-ray field. An image-intensifier tube captures the modulated

X-ray field and converts it to visible light. The resulting visible image is scanned by a camera and then digitized.

Recently, X-ray image *sequences* are used as a diagnostic tool to monitor the physical dynamics for instance of heart and kidneys [103]. To avoid blurring of moving objects, a short exposure time is used. The short exposure time in combination with the requirement to limit X-ray exposure to patient and staff causes the image frames to be quantum limited. These image sequences are therefore corrupted by signal-dependent quantum noise. Although there are many factors in the X-ray imaging chain that affect the overall image quality, this noise is the dominant image corruption associated with reduction in dosage [103, 115].

In earlier work, some methods were developed to filter *single images* degraded by quantum noise [101, 104]. Clinical image *sequence* filtering was investigated in [50]. However, here the images were simulated with white, independent Gaussian noise, which is not justified for simulating quantum noise. In [102, 103] a switching filter was used to process image sequences. Recently, in [4] a maximum-likelihood formulation was employed. We have proposed an OS-supported LLMMSE filter in [116].

6.2.2 Modeling and statistical properties

The passage of photons through the patient under X-ray excitation can be described by an inhomogeneous Poisson counting process [117]. That is, a counting process with a varying mean depending on the properties of the intervening tissue, which changes in place and time. This mean is essentially the diagnostic information in the image sequence. We investigate the statistics of the quantum-limited imaging process to demonstrate the signal-dependent noise and to derive an estimator for the original signal.

Observation model

For quantum-limited imaging, the number of photons captured in each pixel at the camera, appears statistically independent of the other pixels [101, 102, 103, 104]. The *number of photons* captured, expressed by the *counting process* $c(i,j,k)$, can therefore be described as a discrete Poisson random variate:

$$p\{c(i,j,k)|f(i,j,k)\} = \frac{e^{-\lambda f(i,j,k)}(\lambda f(i,j,k))^{c(i,j,k)}}{(c(i,j,k))!}. \tag{6.1}$$

The conditional mean and variance are given by [118]:

$$\mu_{c|f}(i,j,k) = \sigma^2_{c|f}(i,j,k) = \lambda f(i,j,k). \tag{6.2}$$

The parameter λ serves as a known *proportionality factor* relating the displayed image intensity to the number of counts:

$$g(i,j,k) = c(i,j,k)/\lambda. \tag{6.3}$$

This proportionality factor is introduced for convenience. In this way, the diagnostic information $f(i,j,k)$ can also be regarded as an intensity image with the same intensity range as the observed image $g(i,j,k)$.

The conditional mean and variance for $g(i,j,k)$ become, using (6.3):

$$\begin{aligned} \mu_{g|f}(i,j,k) &= f(i,j,k), & (6.4)\\ \sigma^2_{g|f}(i,j,k) &= \frac{f(i,j,k)}{\lambda}. & (6.5) \end{aligned}$$

The properties of the tissue, i.e. the original signal $f(i,j,k)$, are modeled as follows [101]:

$$f(i,j,k) \sim N\{\mu_f(i,j,k), \sigma_f^2(i,j,k)\}, \tag{6.6}$$

i.e. $f(i,j,k)$ is Gaussian distributed with non-stationary mean $\mu_f(i,j,k)$ and non-stationary variance $\sigma_f^2(i,j,k)$.

With these conditional statistics and the statistics of $f(i,j,k)$, the (unconditional) mean of $g(i,j,k)$ can be derived:

$$\mu_g(i,j,k) = E_f\{E_g\{g(i,j,k)|f(i,j,k)\}\} = \mu_f(i,j,k), \tag{6.7}$$

where $E_f\{\}$ and $E_g\{\}$ denote averaging over f and g, respectively. The (unconditional) variance of $g(i,j,k)$ becomes:

$$\begin{aligned} \sigma_g^2(i,j,k) &= E_f\{E_g\{g^2(i,j,k)|f(i,j,k)\}\} - \mu_g^2(i,j,k)\\ &= E_f\{\sigma^2_{g|f}(i,j,k) + \mu^2_{g|f}(i,j,k)\} - \mu_f^2(i,j,k)\\ &= \frac{\mu_f(i,j,k)}{\lambda} + \sigma_f^2(i,j,k). & (6.8) \end{aligned}$$

Statistical properties of the quantum noise

An "additive noise model" for the observation equation can be imposed:

$$g(i,j,k) = f(i,j,k) + n(i,j,k). \tag{6.9}$$

Because the PDF of $g(i,j,k)$ has a scaled Poisson shape, the PDF of the noise $n(i,j,k)$ has a scaled and translated Poisson shape with mean [103]:

$$E\{n(i,j,k)\} = \mu_g(i,j,k) - \mu_f(i,j,k) = 0. \tag{6.10}$$

It can be shown that $n(i,j,k)$ is orthogonal to $f(i,j,k)$:

$$\begin{aligned} E\{n(i,j,k)f(i,j,k)\} &= E_f\{E_n\{n(i,j,k)f(i,j,k)|f(i,j,k)\}\} \\ &= E_f\{f(i,j,k)E_n\{n(i,j,k)|f(i,j,k)\}\} \\ &= E_f\{f(i,j,k)(E_g\{g(i,j,k)|f(i,j,k)\} - f(i,j,k))\} \\ &= E_f\{f(i,j,k)(f(i,j,k) - f(i,j,k))\} = 0. \qquad (6.11) \end{aligned}$$

Although $n(i,j,k)$ and $f(i,j,k)$ are uncorrelated by (6.10) and (6.11), $n(i,j,k)$ is dependent as can be seen from the following expression for the noise variance:

$$\begin{aligned} \sigma_n^2(i,j,k) &= \sigma_g^2(i,j,k) - \sigma_f^2(i,j,k), \\ &= \frac{\mu_f(i,j,k)}{\lambda}. \qquad (6.12) \end{aligned}$$

Equation (6.12) shows that the noise variance *depends* on the mean value of the original signal $f(i,j,k)$. In practice, it can be seen that the noise variance is larger in brighter parts of the image than in the dark areas. The left image from Figure 6.10 can be consulted for this phenomena.

6.2.3 Estimating the modulation process from Poisson observations

In this section, we derive an OS estimator which is designed to estimate the modulation process, $\lambda f(i,j,k)$ from the quantum-limited observations. As we have seen, $\lambda f(i,j,k)$ is the mean of the Poisson-distributed counting process $c(i,j,k)$. We will focus on the estimation of this mean from the counting process, which can be retrieved from $g(i,j,k)$ by inversion of (6.3). First, we will show that the maximum-likelihood estimator of Poisson mean is the sample average.

The maximum-likelihood estimator for Poisson mean

Our aim is to estimate $\lambda f(i,j,k)$ from a sample of the counting process denoted as: $[c_1, c_2, \cdots, c_m]$ which resulted in the current number of photons $c(i,j,k) = \lambda g(i,j,k)$. The simultaneous distribution of this sample, assuming internal independence, is:

$$p\{[c_1, c_2, \cdots, c_m]|\lambda f(i,j,k)\} = \prod_{i=1}^{m} \frac{(\lambda f(i,j,k))^{c_i}}{e^{\lambda f(i,j,k)} c_i!}. \qquad (6.13)$$

This results in the following log-likelihood function:

$$\log\{p\{[c_1, c_2, \cdots, c_m]|\lambda f(i,j,k)\}\} = \sum_{i=1}^{m} -\lambda f(i,j,k) + c_i \log\{\lambda f(i,j,k)\} - \log\{c_i!\}. \qquad (6.14)$$

By maximizing this function with respect to $\lambda f(i,j,k)$ the estimate of $\lambda f(i,j,k)$ can be found:

$$\lambda\hat{f}(i,j,k) = \frac{\sum_{i=1}^{m} c_i}{m}. \tag{6.15}$$

Note that this maximum likelihood is the average of the counts within the sample.

In practical processing of image sequences local ergodicy has to be assumed as the ensemble estimates are exchanged by estimates from spatio-temporal samples. On these spatio-temporal samples, the maximum-likelihood estimator will not be useful. This is because of the dynamic nature of the spatio-temporal original intensity $\lambda f(i,j,k)$ which causes the assumption to fail. The sample average will give inaccurate estimates when the sample does not contain a homogeneous set of counts.

An OS estimator for Poisson mean

By ordering the sample according to value: $[c_{(1)}, c_{(2)}, \cdots, c_{(m)}]$, the structure of the counts can be exploited for robust and efficient estimation purposes [93, 96, 108, 119, 90].

In effect, the rank averages describe the *shape* of the underlying PDF. By using the affine relation between the observation ranks and the rank averages from a normalized "parent" PDF, the mean of the observation PDF can be estimated by applying linear regression as demonstrated in Chapters 4 and 5 [92, 93, 96].

However, matters are more complex with the Poisson PDF. Compared with the estimators derived in the previous chapters, the Poisson PDF changes its shape, not only its position and width, as a function of its mean value. Therefore, the relationship between the rank averages of the observation and the parent is not affine as assumed in (4.11), and can not be as easily exploited as the relationship with most other distributions. We will, however, approximate a linear relation, in order to use the familiar linear-regression technique. In addition, it will be demonstrated that this affine relation suffices for the purpose intended.

We introduce a parent Poisson PDF with known mean value x and rank averages denoted as $p_{(r)}[x]$. Let the averages of the ranks of the observed counting process with mean value $\lambda f(i,j,k)$ be denoted as $E\{c_{(r)}\} = p_{(r)}[\lambda f]$. In order to estimate $\lambda f(i,j,k)$, the following affine relation is "fitted" between the rank averages of the observation and parent PDF:

$$p_{(r)}[\lambda f] = a p_{(r)}[x] + b, \quad \forall\, 1 \leq r \leq m. \tag{6.16}$$

If linear regression is performed on this relation, estimates for a and b are found. In the linear regression, the following problem is solved:

$$a, b \leftarrow \min_{a',b'} \sum_{r=1}^{m} (p_{(r)}[\lambda f] - a' p_{(r)}[x] - b')^2. \tag{6.17}$$

Analytically, this gives the following exact values for a and b:

$$\begin{aligned} a &= \frac{\sum_{r=1}^{m} p_{(r)}[x] p_{(r)}[\lambda f] - \lambda f(i,j,k) \sum_{r=1}^{m} p_{(r)}[x]}{\sum_{r=1}^{m} p_{(r)}[x] p_{(r)}[x] - x \sum_{r=1}^{m} p_{(r)}[x]}, \\ &= \sqrt{\frac{\lambda f(i,j,k)}{x}}; \qquad (6.18) \\ b &= \frac{\sum_{r=1}^{m} p_{(r)}[\lambda f] - a \sum_{r=1}^{m} p_{(r)}[x]}{m}, \\ &= \lambda f(i,j,k) - ax, \\ &= \lambda f(i,j,k) - \sqrt{x \lambda f(i,j,k)}. \qquad (6.19) \end{aligned}$$

In this derivation, the identities $\sum_{r=1}^{m} \mu_r[x] = mx$ and $\sum_{r=1}^{m} \mu_r[\lambda f] = m\lambda f(i,j,k)$ are used [119, 120].

To estimate the current original intensity, the parameters a and b are estimated from the observation using linear regression to solve (6.16). The intensity of the original signal which generated the counting process $c(i,j,k)$ can then be found using the analytical expressions from (6.18) and (6.19) [121]:

$$\lambda \hat{f}(i,j,k) = \hat{a}x + \hat{b}. \tag{6.20}$$

Applying the IWLS estimation procedure

In practice, the rank averages of the observation are not available, and Equation (6.16) is replaced by:

$$c_{(r)} = a p_{(r)}[x] + b + \epsilon_r, \quad \forall\, 1 \leq r \leq m, \tag{6.21}$$

where ϵ_r are the residual errors which are caused by removing the expectation over $c_{(r)}$.

To obtain proper estimation results, the presence of outliers has to be taken into account. In Chapter 5 we considered two robust estimation methods: Iterated Weighed Least Squares (IWLS) and the Range Test (RT). The computationally efficient RT is not directly applicable in this situation because it uses decision thresholds based on the observation PDF which is not traceable because its mean is a stochastic, unknown, value. Therefore, we have relied on the IWLS procedure to solve the minimization problem in a robust way.

This resulted in the following iterative estimation procedure:

$$\begin{bmatrix} \hat{a} \\ \hat{b} \end{bmatrix}^{s+1} \left[\boldsymbol{A}^T \boldsymbol{C}^{-1}_{(yy)} \boldsymbol{W}^s \boldsymbol{A}\right]^{-1} \boldsymbol{A}^T \boldsymbol{C}^{-1}_{(yy)} \boldsymbol{W}^s \begin{bmatrix} c_{(1)} \\ \vdots \\ c_{(m)} \end{bmatrix}. \tag{6.22}$$

In the present situation, the design matrix $\boldsymbol{A}$ is composed as follows:

$$\boldsymbol{A}^T = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \mu_1[x] & \mu_2[x] & \cdots & \mu_m[x] \end{bmatrix}. \tag{6.23}$$

Initial estimates and the choice for x

Robust initial estimates to start the iteration procedure are:

$$\hat{a}^0 = \sqrt{\frac{c(i,j,k)}{x}} \quad \text{and} \quad \hat{b}^0 = c(i,j,k) - \sqrt{c(i,j,k)x}, \tag{6.24}$$

where the current count $c(i,j,k)$ is used. The iteration procedure can be stopped if $\hat{a}$ and $\hat{b}$ have converged, which is usually within a few steps.

The parameter x describes the shape of the parent PDF. In effect, it determines the entries of $\boldsymbol{C}_{(yy)}$ and $\boldsymbol{A}$. In order to achieve a low estimation variance, the PDF shapes have to be comparable, which is guaranteed if $x \approx c(i,j,k)$. For each sample the matrices $\boldsymbol{C}_{(yy)}$ and $\boldsymbol{A}$ have to be built, which is not a heavy burden as they can be stored efficiently for several values of x or calculated in an analytical way. In addition, the estimation process is only sensitive to values of x for the lower region of counts $c(i,j,k) < 10$. This gives a considerable degree of freedom and reduced complexity in choosing x for higher counts.

Analysis of the affine approximation

The linear relation in (6.16) is sufficient for the purpose of estimating the mean of an observation given a known parent. This can be seen by looking at the Taylor approximation of $p_{\lambda f}\{\}$, a Poisson PDF with mean λf, by $p_x\{\}$, a PDF with intensity x. The k^{th} order approximation of $p_{\lambda f}\{c\}$ by $p_x\{c\}$ is given by:

$$p_{\lambda f}\{c\} \approx \sum_{j=0}^{k} \frac{(\lambda f - x)^j}{j!} \sum_{i=0}^{j} \binom{j}{i} (-1)^{j-i} p_x\{c-i\}. \tag{6.25}$$

The average value of this approximation is given by:

$$E_c\left\{\sum_{j=0}^{k} \frac{(\lambda f - x)^j}{j!} \sum_{i=0}^{j} \binom{j}{i} (-1)^{j-i} p_x\{c-i\}\right\} \tag{6.26}$$

$$= \sum_{j=0}^{k} \frac{(\lambda f - x)^j}{j!} \sum_{i=0}^{j} \binom{j}{i} (-1)^{j-i} \sum_{c=1}^{\infty} c p_x\{c-i\}, \tag{6.27}$$

which is equal to:

$$\sum_{j=0}^{k} \frac{(\lambda f - x)^j}{j!} \sum_{i=0}^{j} \binom{j}{i} (-1)^{j-i} (x+i) \tag{6.28}$$

$$= \sum_{j=0}^{k} \frac{(\lambda f - x)^j}{j!} \left\{ x \sum_{i=0}^{j} \binom{j}{i} (-1)^{j-i} + j \sum_{i=0}^{j} \binom{j-1}{i-1} (-1)^{j-i} \right\}, \tag{6.29}$$

$$= \sum_{j=0}^{k} \frac{(\lambda f - x)^j}{j!} \{x(0)^j + j(0)^{j-1}\} = \lambda f. \tag{6.30}$$

It appears that the last summation is only nonzero for $j = 0$ and $j = 1$ which means that the average λf can be sufficiently estimated from a *linear* approximation of $p_{\lambda f}\{\}$ by $p_x\{\}$.

6.3 Gamma-corrected video signals

Most video sequences that we regard as "original" data are corrupted by a certain amount of noise originating from the camera electronics. From measurements, the SNR of un-processed image sequences appears to be unexpectedly low. For instance, the well-known "mobile" sequence has an estimated SNR of 28dB.

The noise in "original" signals originates as thermal noise from the early electronic circuits of the camera. The captured (noisy) signal passes a gamma-correction stage that compensates for the non-linearity of a Cathode Ray Tube (CRT). An effect of this gamma correction is that the noise in the observed "original" sequence has statistics that *depend* on the captured intensity.

In this section we will consider a method that is able to effectively reduce the noise in the gamma-corrected "original" sequences. The method includes an order-statistic estimator with weights that are optimally adapted, given the observation model and PDF of the noise. The resulting filter is an efficient OS-FIR filter (see Chapter 2) that can be used to reduce the noise in "original" video sequences.

6.3.1 Signal model

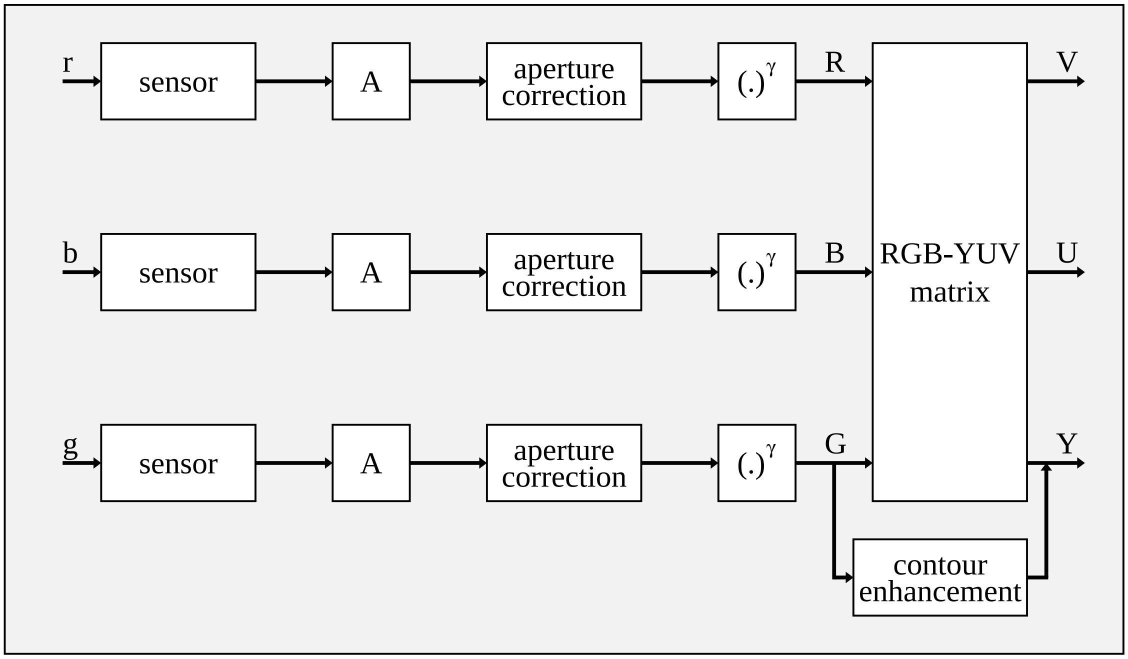


FIGURE 6.2: THE SCHEMATIC STRUCTURE OF A COLOR VIDEO CAMERA

The structure of a color camera

The properties of camera noise intrinsically depend on the camera structure. This structure can be schematically modelled for a color camera as in Figure 6.2. It consists of three recognizable paths relating to the three colors red (R), green (G), and blue (B).

The electrical current generated by the light sensors depends directly on the object intensity, and is amplified by an electronic amplifier (A) to a suitable level. After amplification, the smoothing effects caused by the finite area of the sensor elements are reduced in the aperture-correction stage. This stage usually consists of spatial high-emphasis filters.

To match the characteristics of a CRT, the video signal passes a gamma-correction stage. Finally, in a processing stage, known as the "RGB-YUV matrix", the R, G, and B signals are transformed to the YUV system. In this system Y is the intensity signal and U and V comprise the color information. A portion of the G signal, which usually contributes most to the intensity signal Y, is forwarded to the Y path for spatial sharpening purposes. This "crispening" action and the aperture correction can be switched off on high-quality cameras, which facilitates additional signal processing.

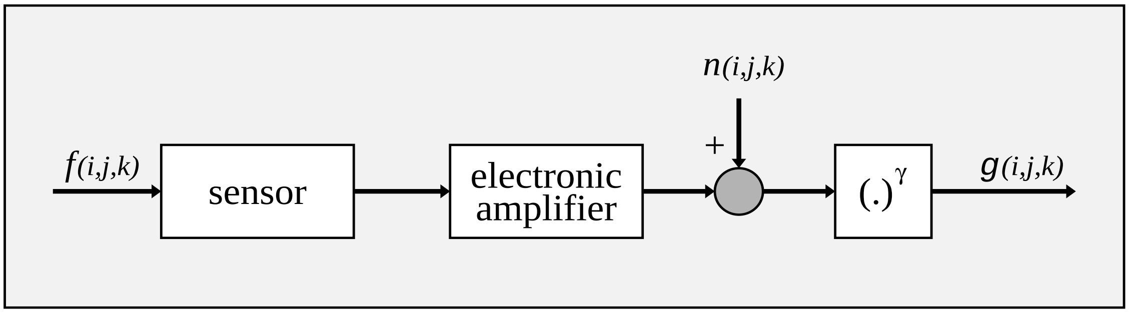


Figure 6.3: A schematic model of the path that object intensities follow yielding a pixel value

Statistical properties of the gamma-correction

For the purpose of investigating noise filtering of gamma-corrected video signals we have simplified the camera structure to the signal path shown in Figure 6.3. In this simplified structure we have focussed on one color signal which is regarded as a common intensity signal. We have ignored the aperture correction and crispening filter. The signal path now begins with amplifying the input signal $f(i, j, k)$. The thermal noise introduced by the amplifier can be modeled as zero-mean, signal-independent, white Gaussian noise [122, 123]. The noise variance σ_n^2 is time invariant and is assumed to be known.

Ignoring the amplification factor for simplicity, the corrupted signal can be written as $f(i, j, k) + n(i, j, k)$. After the gamma correction, the observed signal $g(i, j, k)$ is modeled as [123]:

$$g(i,j,k) = (f(i,j,k) + n(i,j,k))^{\gamma} \; = f^{\gamma}(i,j,k)\left(1 + \frac{n(i,j,k)}{f(i,j,k)}\right)^{\gamma}. \tag{6.31}$$

with a typical value of $\gamma = 0.45$. Note that we have assumed in this equation that image intensities lie between 0 and 1.

Some examples of the influence of gamma correction on images are illustrated in Figure 6.4. The top image has a $\gamma < 1$ and appears lighter; the darker parts of the image are emphasized. The bottom image has a $\gamma > 1$ and appears darker; the lighter parts of the image are emphasized. Also, the input-output relations are shown.

To facilitate noise filtering we approximate (6.31) by a Taylor expansion of order s about $\frac{n(i,j,k)}{f(i,j,k)} = 0$.

$$(f(i,j,k) + n(i,j,k))^{\gamma} \approx f^{\gamma}(i,j,k)\left(1 + \sum_{l=1}^{s} \binom{\gamma}{l} \frac{n^{l}(i,j,k)}{f^{l}(i,j,k)}\right), \tag{6.32}$$

where we have used the convenient short-hand notation:

$$\binom{\gamma}{l} = \frac{\gamma(\gamma-1)\cdots(\gamma-l+1)}{l!}. \tag{6.33}$$

The above approximation now enables us to rewrite the observation equation as an additive noise model:

$$\begin{aligned} g(i,j,k) &\approx f^{\gamma}(i,j,k) + n_{\gamma}(i,j,k); \\ n_{\gamma}(i,j,k) &= f^{\gamma}(i,j,k)\sum_{l=1}^{s}\binom{\gamma}{l}\frac{n^{l}(i,j,k)}{f^{l}(i,j,k)}. \end{aligned} \tag{6.34}$$

For practical values of $f(i,j,k)$ and $n(i,j,k)$, the approximation is acceptable if $s > 1$. We have used $s = 4$ to guarantee sufficient accuracy. A 1^{st}-order Taylor expansion is not accurate enough at low frequencies [124].

The additive noise component is signal dependent. This can be shown after specifying the stochastic models for the original intensity $f(i,j,k)$ and the additive noise $n(i,j,k)$:

$$f(i,j,k) \sim N\{f(i,j,k), 0\}; \quad n(i,j,k) \sim N\{0, \sigma_n^2\}. \tag{6.35}$$

Using the additional properties $E\{n^3(i,j,k)\} = 0$ and $E\{n^4(i,j,k)\} = 3\sigma_n^2$ for the Gaussian noise, the average value of the observation noise is approximated by ($s = 4$):

$$\mu_{n_\gamma}(i,j,k) \approx \left(\binom{\gamma}{2} f^{\gamma-2}(i,j,k) + 3\binom{\gamma}{4} f^{\gamma-4}(i,j,k)\right)\sigma_n^2. \tag{6.36}$$

The noise variance can be approximated for moderate and larger values of $f(i,j,k)$ as [123]:

$$\sigma_{n_\gamma}^2(i,j,k) \approx \gamma^2\sigma_n^2 f^{2(1-\gamma)}(i,j,k) - \mu_{n_\gamma}^2(i,j,k). \tag{6.37}$$

It can be seen that the noise $n_\gamma(i,j,k)$ is not zero-mean and signal dependent as the statistics depend on the value of the original signal $f(i,j,k)$. Note that if $\gamma = 1$, the noise becomes zero-mean and the dependency disappears.

6.3.2 An order-statistic filter for γ-corrected noisy signals

The desired signal is $f^{\gamma}(i,j,k)$, and has to be estimated from $g(i,j,k)$. An obvious approach would be to create a transformed observation $g'(i,j,k)$ by inversion of the gamma correction. The signal $g'(i,j,k)$ can then be filtered and gamma-corrected to establish the final result. However, this approach is hindered by the presence of (additive) quantization noise on the digitized signal $g(i,j,k)$. In addition, noticable aliasing effects might be introduced.

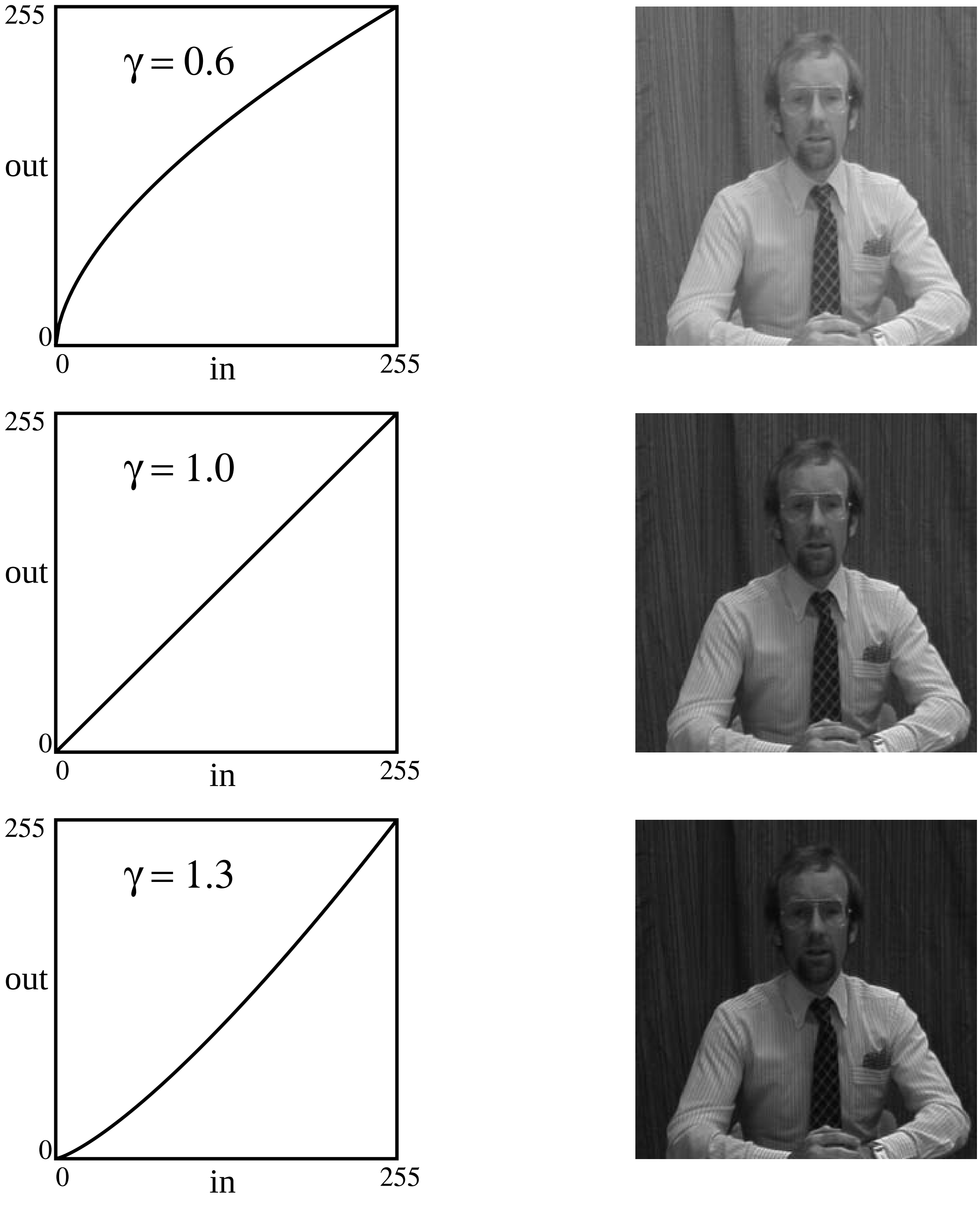


Figure 6.4: The effect of gamma correction on images, illustrated by the intensity-transfer curves and gray-value images

Most published noise filters for video sequences do not take the gamma correction into account. As a result, the sequence will have a darker appearance after processing. We propose a spatio-temporal OS-FIR filter that does take the correction into account [124]. It is again based on an OS estimator:

$$\hat{f}^{\gamma}(i,j,k) = \sum_{r=1}^{m} w_r g_{(r)}(i,j,k). \tag{6.38}$$

Here, $g_{(r)}(i,j,k)$ are the ordered observations within a spatio-temporal window of size m, centered at (i,j,k). The weights w_r are fixed values that are optimal for estimating a constant signal in signal-dependent noise. The weights depend on the PDF of the noise $n(i,j,k)$ in (6.31). On the basis of (6.34) and the signal and noise properties in (6.35), we find the following expectation for $g(i,j,k)$:

$$E\{g(i,j,k)\} = f^{\gamma}(i,j,k) + f^{\gamma}(i,j,k)\sum_{l=1}^{s}\binom{\gamma}{l}\frac{E\{n^l(i,j,k)\}}{f^l(i,j,k)}. \tag{6.39}$$

The above relation still holds after ordering the data in m ranks. Thus we can write for every rank $1 \leq r \leq m$ of $g(i,j,k)$:

$$E\{g_{(r)}(i,j,k)\} = f^{\gamma}(i,j,k) + f^{\gamma}(i,j,k)\sum_{l=1}^{s}\binom{\gamma}{l}\frac{E\{n^l_{(r)}(i,j,k)\}}{f^l(i,j,k)}. \tag{6.40}$$

This equation includes the expectation of the ranks taken to the power l, i.e. the higher-order moments of ordered data. These moments, $E\{n^l_{(r)}(i,j,k)\}$, are readily available if the statistics of the noise source are known. However, the expectations over the ordered observations $E\{g_{(r)}(i,j,k)\}$ are not available. By approximating these expectations by the observed ordered observations themselves, we arrive at:

$$g_{(r)}(i,j,k) = f^{\gamma}(i,j,k) + f^{\gamma}(i,j,k)\sum_{l=1}^{s}\binom{\gamma}{l}\frac{E\{n^l_{(r)}(i,j,k)\}}{f^l(i,j,k)} + \epsilon_r(i,j,k), \tag{6.41}$$

where $\epsilon_r(i,j,k)$ is a stochastic error term due to approximating the expectations.

The relations in (6.41) can be written in a matrix-vector form as follows:

$$\begin{bmatrix} g_{(1)}(i,j,k) \\ \vdots \\ g_{(m)}(i,j,k) \end{bmatrix} = \begin{bmatrix} 1 & \cdots & E\{n^s_{(1)}(i,j,k)\} \\ \vdots & \vdots & \vdots \\ 1 & \cdots & E\{n^s_{(m)}(i,j,k)\} \end{bmatrix} \begin{bmatrix} f^{\gamma}(i,j,k) \\ \vdots \\ \binom{\gamma}{s} f^{\gamma-s}(i,j,k) \end{bmatrix} + \begin{bmatrix} \epsilon_1(i,j,k) \\ \vdots \\ \epsilon_m(i,j,k) \end{bmatrix}. \tag{6.42}$$

Or, using a matrix-vector notation:

$$\boldsymbol{g}_{()}(i,j,k) = \boldsymbol{A}\boldsymbol{\theta}(i,j,k) + \boldsymbol{\epsilon}(i,j,k). \tag{6.43}$$

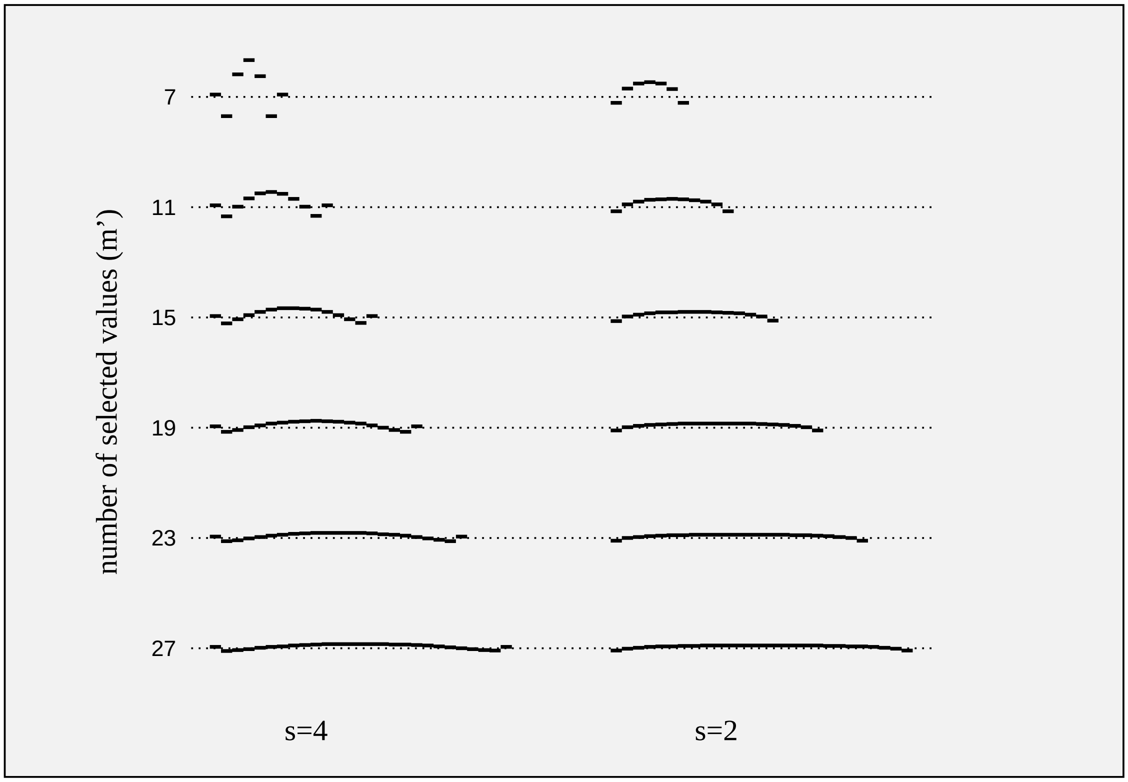


Figure 6.5: A graphical illustration of the OS-FIR filter shapes for several window sizes and for 2^{nd} and 4^{th} order approximation

Since $E\{n^l_{(r)}(i,j,k)\}$, for $1 \leq l \leq s$, are known constants, the matrix $\boldsymbol{A}$ is entirely known. Equations (6.42) and (6.43) describe a set of linear relations in the $s+1$ unknown parameters $\binom{\gamma}{l} f^{\gamma-l}(i,j,k)$, for $0 \leq l \leq s$. The top entry of $\boldsymbol{\theta}(i,j,k)$ is $f^{\gamma}(i,j,k)$, the desired signal value.

Usually, for common window sizes $m > s+1$, which means that this set is over-determined. Equation (6.43) is solved in minimum-variance sense of $\boldsymbol{\epsilon}(i,j,k)$ by:

$$\hat{\boldsymbol{\theta}}(i,j,k) = \boldsymbol{A}^{+}\boldsymbol{g}_{()}(i,j,k). \tag{6.44}$$

The pseudo-inverse $\boldsymbol{A}^{+}$ is found by singular value decomposition [125]. The top row of $\boldsymbol{A}^{+}$ contains the weights of the desired OS-FIR filter to estimate $f^{\gamma}(i,j,k)$ from $g(i,j,k)$. This means that only this row has to be computed. In addition, because the PDF of the noise is known from Equation (6.35), the rank moments of the noise establishing $\boldsymbol{A}$ can be calculated in advance. This means that the top row of $\boldsymbol{A}^{+}$ consists of fixed weights for the entire signal.

The filter weights are illustrated in Figure 6.5 for some window sizes between 7 and 27 and for $s=4$ and $s=2$. It can be seen that the use of higher-order rank moments for the 4^{th} order approximation has resulted in more refined filter shapes compared to those resulting from a 2^{nd} order approximation. From experiments it appeared that the filter shapes stay approximately the same if s is increased even further. Note that the approximation order does not directly influence the calculation effort in filtering. A large s has only consequences in the pre-processing step where the weights are determined and on the amount of over-determination of the set of equations.

6.3.3 Robust implementation using the Range Test

The data for the OS filter (6.38) is gathered from a spatio-temporal window centered at (i, j, k). Because the value of every pixel used in filtering has an influence on the final result, the OS filter designed so far is not robust. A selection procedure to define the homogeneous part of the operation window has to ascertain the absence of outlying observations.

Using the model information, we can approximate the PDF of the observed signal:

$$g^{1/\gamma}(i, j, k) \sim N\{f(i, j, k), \sigma_n^2\}. \tag{6.45}$$

This approximation and the assumptions regarding the distribution of $f(i, j, k)$:

$$f(i, j, k) \sim N\{f(i, j, k), 0\}, \tag{6.46}$$

enable us to use the Range Test. This test was described in Chapter 5 as the heart of a detecting and selecting stage.

The result of the Range Test is an ordered homogeneous selection of pixel values comprising the current pixel. Based on the number of pixels in this selection, a number of filter weights has to be calculated. Because the set of selectable sizes is limited and known, it is possible to pre-compute and store the filter weights for all possible sizes. The overall structure of the filter can now be seen from Figure 6.6. In this figure, we can distinguish the ordering stage, the Range Test which uses a decision table to evaluate pixel values, and the filtering stage. The latter stage consists of a bank of OS-FIR filters with fixed weights. The appropriate filter is chosen on basis of the number of homogeneous pixels that were found using the range test.

6.4 Experimental evaluation

6.4.1 Quantum-limited sequences

We have applied the OS estimator for quantum-limited data on synthetically distorted sequences and on a clinical sequence suffering from quantum noise. The

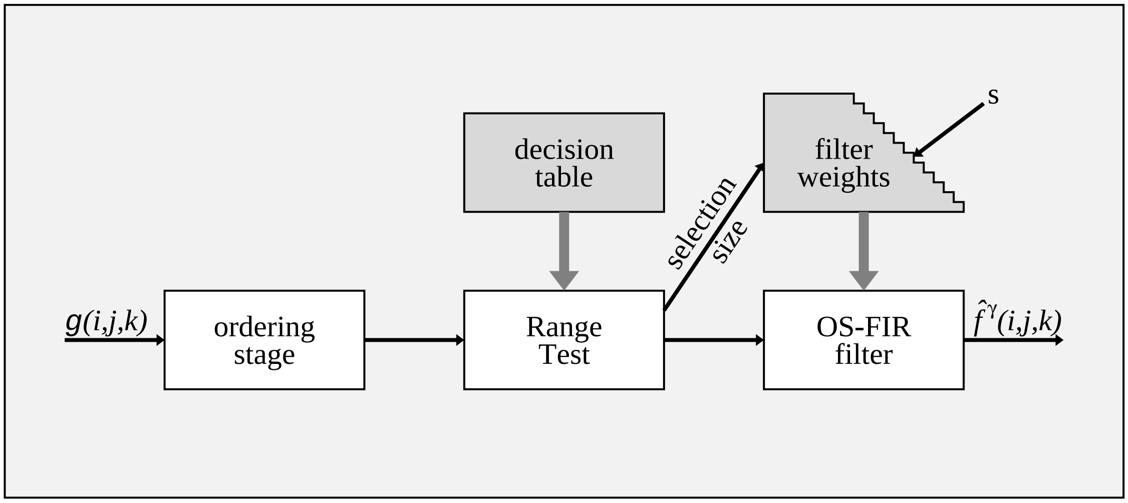


Figure 6.6: Overview of the proposed robust OS-FIR filter for gamma-corrected video signals

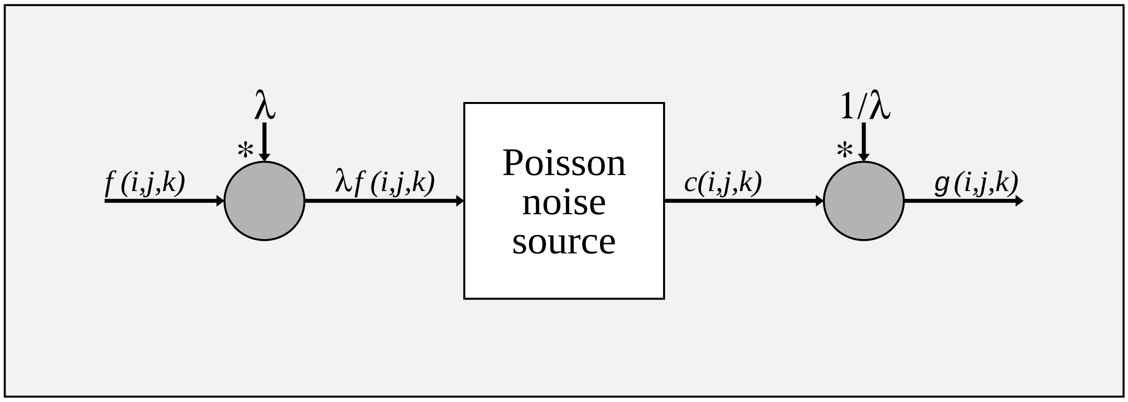


Figure 6.7: The simulation method used for introducing quantum noise in sequences

mechanism to introduce synthetic quantum noise in regular sequences is shown in Figure 6.7. First, all pixels values are multiplied by the proportionality factor λ. The resultant values are then used as the mean value for the Poisson noise source, yielding the counting process. To retrieve image intensities $g(i,j,k)$, comparable with $f(i,j,k)$, the counts are divided by λ [103, 126]. It is clear that the value of λ necessary to scale $c(i,j,k)$ to common image intensities reflects the SNR of $g(i,j,k)$. A large λ indicates a high SNR whereas a low λ indicates a low SNR.

We have synthetically distorted the "mobile" sequence using the simulation model above for $\lambda = 0.5$, 0.25 and 0.1. Expressed as SNRs (calculated in the usual manner between $f(i,j,k)$ and $g(i,j,k)$), these values amount to respectively 12, 9 and 5dB. The effect of scaling with λ can be clearly seen from the histograms of the resulting displayed data. For a comparison, the histograms of the original "mobile" sequence and corrupted sequences with $\lambda = 0.5,\ 0.25,\ 0.1$ are shown in Figure 6.8. For lower

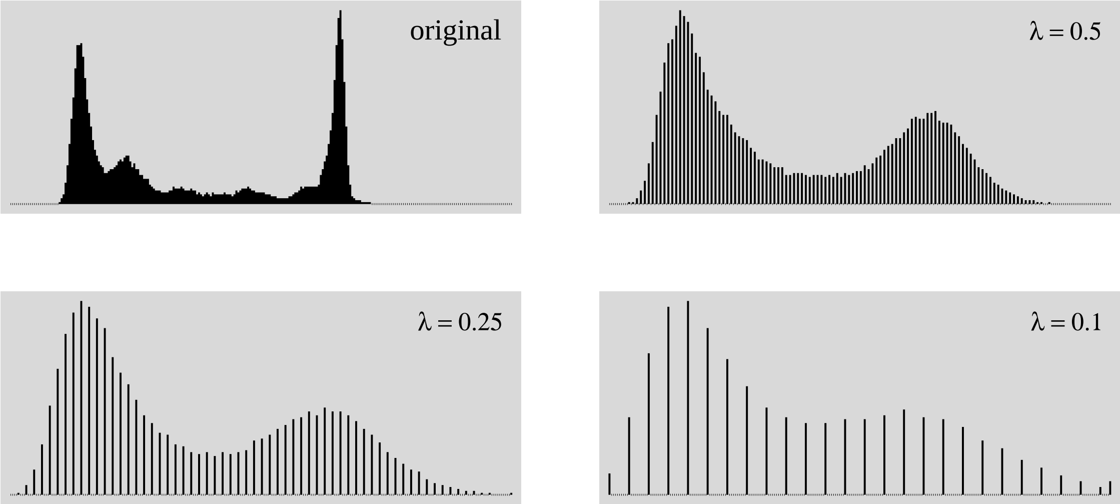


Figure 6.8: The histogram of the original sequence and of the corrupted image sequences for various values of the proportionality factor λ

Table 6.1: *Average Improvement in* SNR for the "mobile" sequence.

λ	SNR (dB)	SNRi (range)
0.5	12.	6.9 (0.3)
0.25	9.	7.2 (0.4)
0.1	5.	9.7 (0.4)

values of λ, fewer numbers of gray-level remain in the image. This "quantization" effect can be used to identify λ in practical situations of interest.

The OS estimator was used with a spatio-temporal symmetrical window of $3 \times 3 \times 3$ pixels. The original intensity was estimated by the robust IWLS procedure. Motion compensation was not used for two reasons. First, our recursive motion estimator from Section 3.4.2 was unable to track the irregular motion that is usually present in clinical sequences. Second, the motion estimator is optimized for Gaussian noise and will therefore give sub-optimal results in the presence of asymmetric Poisson noise [103].

The SNR and the average improvement for the synthetic data is shown in Table 6.1. Frames of the distorted and filtered sequence are shown in Figure 6.9 for $\lambda = 0.5$, in Figure 6.10 for $\lambda = 0.25$ and in Figure 6.11 for $\lambda = 0.1$, respectively.

To illustrate the performance of the OS estimator on genuine quantum-limited data, we have filtered a clinical image sequence of a cardiac scene showing some blood vessels and tissue. This clinical sequence suffers from quantum noise with an estimated $\lambda = 0.5$. An image of the observed sequence and the filter result are

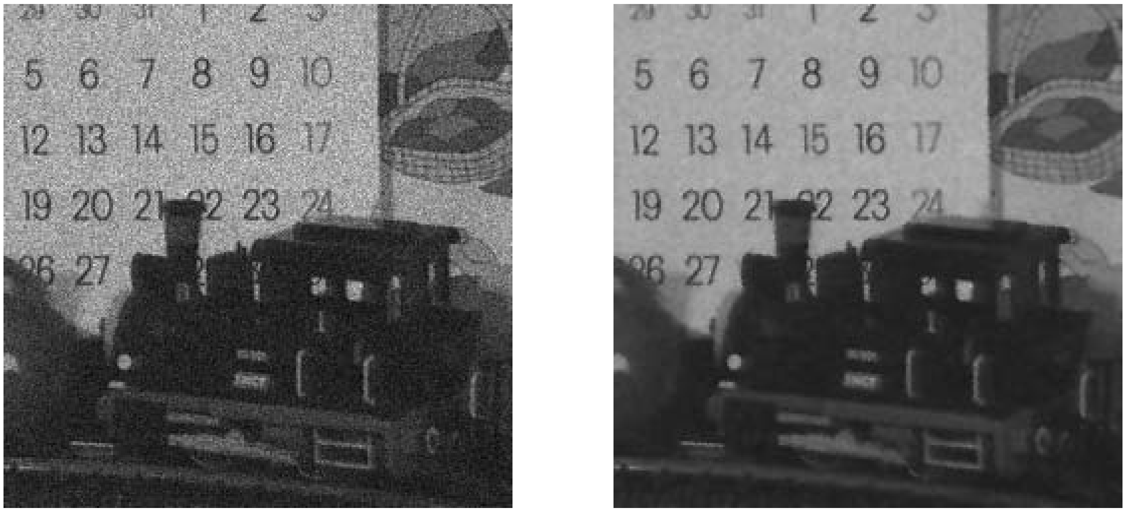


Figure 6.9: Observed and filtered frame for $\lambda = 0.5$

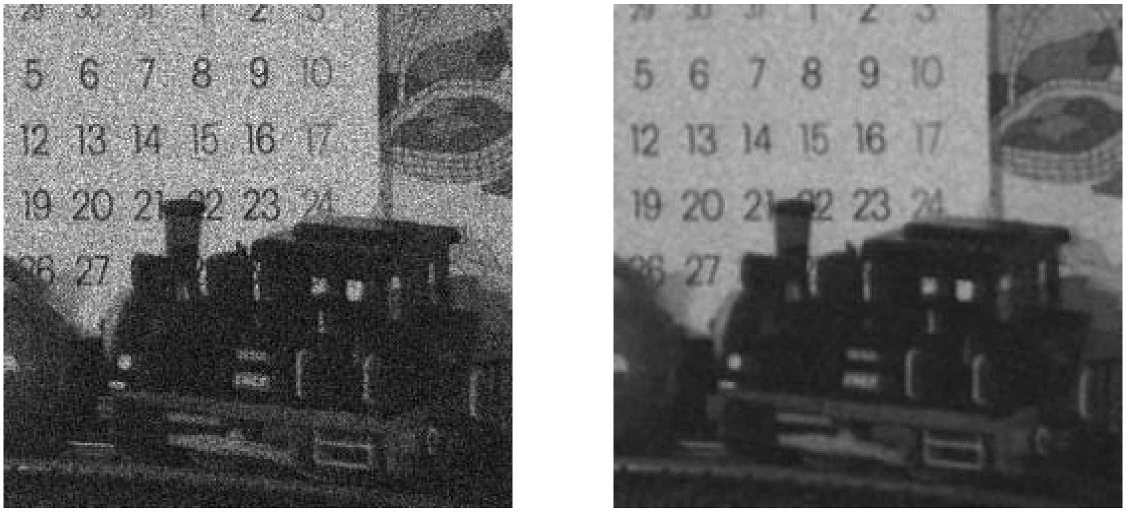


Figure 6.10: Observed and filtered frame for $\lambda = 0.25$

shown in Figure 6.12. As the original signal is not available, the improvement can only be evaluated from a visual point of view. It appears that the noise is reduced without affecting the sharpness.

6.4.2 Gamma-corrected video signals

In the experimental evaluation of the OS-FIR filter for γ-corrected video signals we have considered synthetically distorted sequences and a genuine γ-corrected "original" image sequence which suffers from camera noise. The simulation procedure to create the synthetically distorted sequences and to evaluate the filter improvement is shown in Figure 6.13. Signal-independent Gaussian noise is added to a sequence

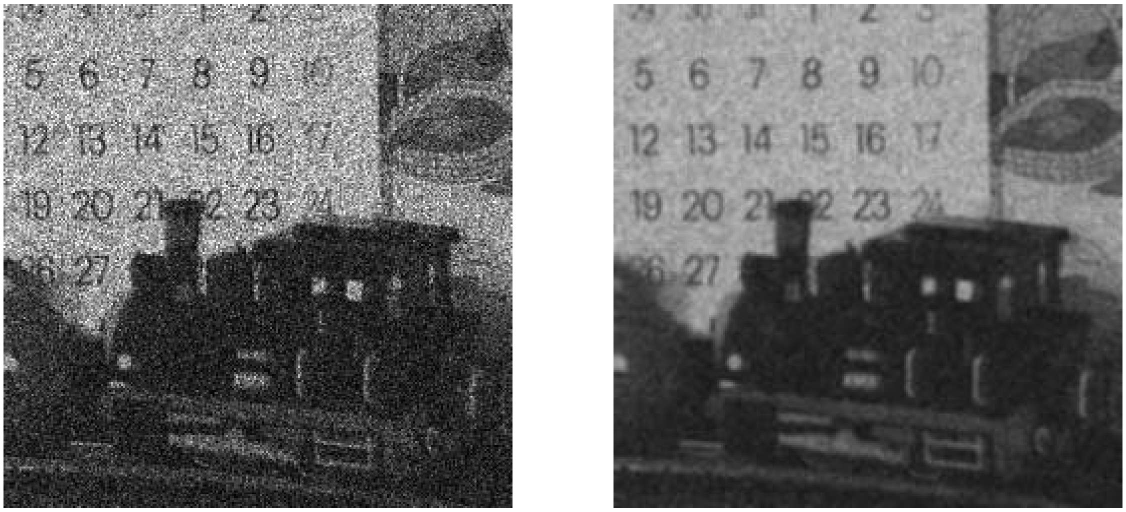


Figure 6.11: Observed and filtered frame for $\lambda = 0.1$

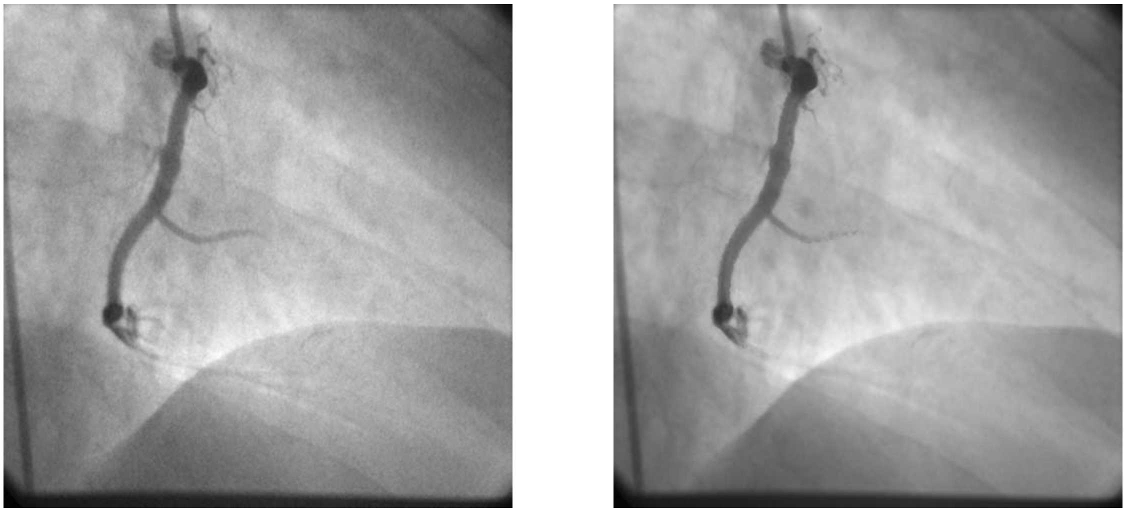


Figure 6.12: A radiographic sequence suffering from noise and the filter output

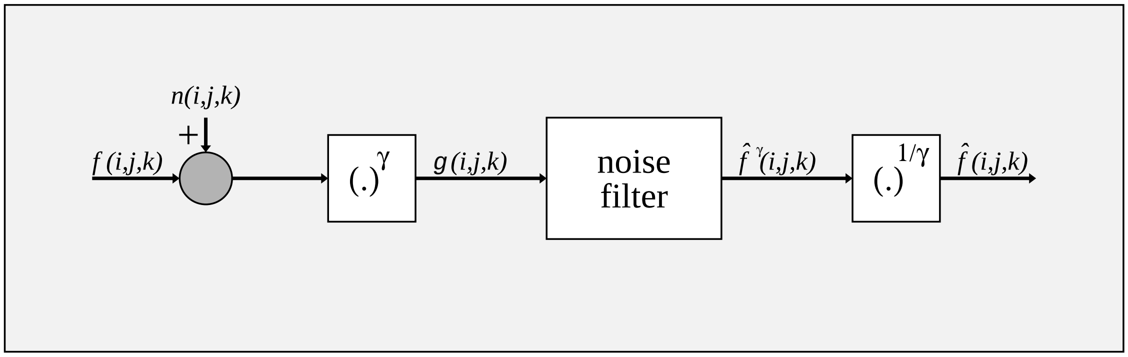


Figure 6.13: The simulation of noisy γ-corrected image sequences

Table 6.2: *Average Improvement in* SNR, (range) for the "mobile" sequence at 10dB

γ	*Average Improvement in* SNR (range)			
	s=1	s=2	s=3	s=4
0.35	7.14 (0.2)	7.69 (0.2)	7.78 (0.2)	7.78 (0.2)
0.45	7.69 (0.2)	7.78 (0.2)	7.87 (0.2)	7.87 (0.2)
0.55	7.73 (0.2)	7.82 (0.2)	7.88 (0.2)	7.88 (0.2)

and the resulting value undergoes a γ-correction. The real-valued signals are then filtered using the proposed algorithm and subsequently pass a stage where the γ-correction is inverted. This is then regarded as the final result, which is compared with the original sequence to determine the SNRi. Notice that we have neglected the introduction of aliasing effects. The synthetic noise is added to a level of 10dB, which is several orders of magnitude greater in power than the γ-corrected noise present in this "original" sequence. The OS-FIR filter was used for filtering in combination with the Range Test to detect outlying pixel values from a symmetric motion-compensated spatio-temporal window of 27 pixels.

The improvements in SNR of $\hat{f}(i,j,k)$ compared with $f(i,j,k)$ (see Figure 6.13) are given for several filter orders s in Table 6.2. It can be seen that, for $s > 1$, the filter result is practically insensitive to γ. In addition, the filter results are on par with those of the previous chapters. The filters with $s = 1$ are similar to ignoring the gamma correction and effectively mean spatio-temporal averaging supported by the Range Test. From the table it is clear that this gives lower results. It can be seen that the improvement increases slightly for higher filter order s. The improvement has stabilized for $s > 2$.

The filter for γ-corrected video signals is also applied on the original "mobile" sequence. This sequence inherently contains some camera noise which results in an estimated SNR (dynamic range) of around 28dB for the original sequence. This

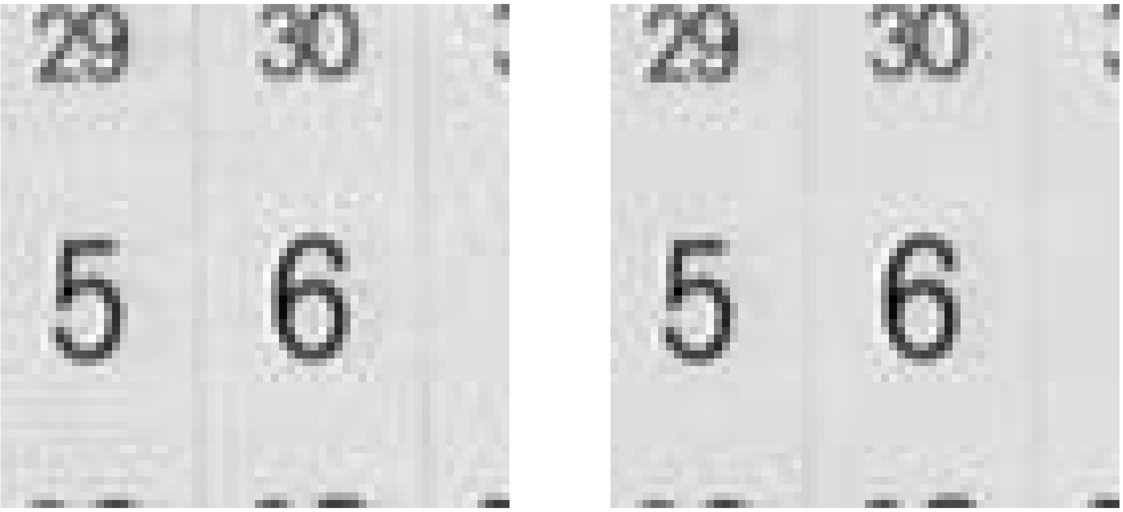


Figure 6.14: The original "mobile" sequence seems to suffer from camera noise, on the left an enlarged and enhanced portion of the image and on the right the filter result

camera noise can be made more visible by enlarging a part of the image and enhancing the contrast (left image of Figure 6.14). The right image in this figure is the result after filtering with an assumption that $\gamma = 0.45$. After filtering, the estimated SNR of the entire sequence becomes 35dB. The resulting image sequence remains reasonably sharp and the noise is reduced.

Chapter 7

Conclusions and topics for further research

This thesis is devoted to noise filtering of image sequences. We have reviewed most well-known techniques in Chapters 2 and 3 and have proposed a number of new techniques in Chapters 4 and 5. In Chapter 6 we have focussed on specific applications. In most chapters, we have performed an experimental evaluation to support and discuss our ideas. This chapter summarizes the conclusions that were reached in these situations. In addition, we will give our view of the path that future research might follow in the field of noise filtering of image sequences.

7.1 Summary of the conclusions arrived at

The main problem in image-sequence noise filtering is the dynamic behavior of the spatio-temporal signal. This is not so much different from the problems encountered in spatial noise filtering, but image sequences require additional filter properties. Non-adaptive filters such as straight averaging do not produce useful results, as moving objects and spatial detail are blurred. Even filters that adapt globally to the signal statistics such as Wiener, Kalman and switching filters with fixed control parameter are unable to give worthwhile results. Temporal median filters remove certain object features while the feature-preserving spatio-temporal median filters appear to be not very noise suppressive.

From the experiments we concluded that only locally-adapting filters are able to give good noise suppression without blurring the signal. Among these filters are some weighed averaging and switching filters, the simultaneous approaches and the adaptive order-statistic filters, as developed in Chapters 4 to 6.

Motion compensation has been widely used as a preprocessing stage to aid the noise filter by supplying homogeneous temporal signals. However, the estimation of motion suffers from the observation noise and the incompleteness of the motion model used. For these reasons, the use of motion compensation will not always result in homogeneous temporal signals that can be filtered by non-adaptive filters. Because of failure of the motion compensation, locally-adapting filters are still necessary for useful results. Motion compensation can only be seen as a first, but not perfect, aid for creating stationary temporal signals.

Spatio-temporal methods are able to give more noise suppression than temporal methods. This is obvious considering the increased amount of data involved with filtering. However, as spatial signals are usually also non-stationary, adaptation is required to avoid blurring. From our experiments we have found that for the proposed order-statistic filters a centered spatio-temporal cube containing 27 pixels gives useful results.

The methods that we have developed in this thesis are spatio-temporal locally-adapting filters. The algorithms use estimates of the local statistics to normalize the non-stationary signal (Chapter 4) or to adapt the filter characteristics locally (Chapters 5 and 6). For estimation of the local statistics we have relied on efficient, in estimation-variance sense, order-statistic estimators. Apart from providing local adaptation in non-stationary situations, the estimators also provide the opportunity to track slowly changing noise statistics.

The adaptation power of the order-statistic estimator is substantially improved by rejecting outlying observations from the data window. For this task we have included the IWLS and the RT procedures. The RT procedure uses the assumption that the original signal is constant within the data window for outlier selection, whereas the IWLS method uses a stochastic model for the original signal. These different assumptions result in slightly better results for the IWLS method in low-noise situations where the constant-intensity assumption fails.

The estimation properties of the order-statistic estimator were shown in applications to imaging processes with signal-dependent noise. These applications were noise-filtering quantum-limited clinical X-ray sequences and gamma-corrected video signals. By linearizing the observation models we have developed dedicated order-statistic estimators. Here, the estimators were used to provide direct intensity estimates instead of estimates of the local statistics. Because of the outlier rejection and the efficient estimation procedure, the filter results are good without unreasonable blurring.

7.2 Suggestions for further research

Considering the conclusions reached, it is clear that the investigations into noise-filter structures are maturing. At this moment, only one filtering structure, the simultaneous motion estimation and filtering approach seems to gain interest in literature. However, the high additional computational effort may limit the practical application of this technique. Our view is that the research in simultaneous motion estimation and filtering will therefore be mainly dedicated to understanding the relational and fundamental concepts of motion estimation and filtering. The future research in image sequence noise filtering will most probably be systems oriented and devoted to specific problems and applications.

The systems orientation is necessary to solve several problems concerning noise filtering in specific environments. As an example, we consider noise filtering for the television environment. Here, the environment influences the choice of the operation window because of interlaced video formats. Also, the aperture correction, gamma correction and crispening filters will influence the filter operation. In this context the noise filtering of color sequences will be interesting.

Noise filtering of image sequences can be investigated in combination with specific problems. For instance, the identification and removal of distortion in combination with noise filtering. This might comprise multi-frame approaches extended with ideas presented in this thesis. A specific application are sequences produced by a surveillance camera that suffer from out of focus and motion blur in addition to the sensor noise.

Another interesting problem where noise filtering techniques can be applied is resolution enhancement of images. In practical situations, time-consecutive recordings of a dynamic scene are available or can be made available. From this image sequence a single image of higher resolution can be retrieved using aliasing effects and fractional motion estimates [127, 128]. However, these resolution enhancement methods strongly suffer from observation noise which is always present in the practical applications. The noise-filtering techniques from this thesis might help to reach better results in practical situations.

Some subjects for further research directly connected with the material of this thesis are estimation procedures for the variance of the observation noise and control of the effect of scene changes and illumination changes on the motion estimation and filter behavior. Other subjects are the use of the developed filters to color image sequences and to sequences in interlaced video format. A very interesting subject, initiated by this thesis, is the development of adaptive order-statistic estimators for filtering of image sequences with complex observation models. We have shown in this thesis that the class of order-statistic estimators has some very interesting properties for adaptive noise filtering. These properties should be exploited!

Appendix A

Stochastic properties of ordered statistics

In this appendix we derive analytical expressions for the distribution functions of a rank (Section A.1), the first and second moments (Section A.2), and the joint distribution functions and correlations (Section A.3). The results hold for both continuous and discrete variates. Finally, in Section A.4, as an example, we will look at two particular distributions, namely the uniform and Gaussian Probability Density Function (PDF).

A.1 Distribution functions of ranks

If we denote an (unordered) sample $\boldsymbol{X}$ of size m by:

$$\boldsymbol{X} = [x_1, x_2, \ldots, x_m]^T, \tag{A.1}$$

and assume that each particular value x_i is an independent variate with Cumulative Distribution Function (CDF) $P(v) = Pr\{x_i \leq v\}$, where $Pr\{\}$ denotes the probability of an event. Then, if the sample is ordered as:

$$\boldsymbol{X} = [x_{(1)}, x_{(2)}, \ldots, x_{(m)}]^T, \tag{A.2}$$

where $x_{(1)} \leq x_{(2)} \leq \ldots \leq x_{(m)}$, the CDF of the largest rank $x_{(m)}$ is given by:

$$R_{(m)}(v) = Pr\{x_{(m)} \leq v\} = Pr\{\text{all } x_i \leq v\} = P^m(v). \tag{A.3}$$

In the same way, the CDF of the smallest rank is:

$$\begin{aligned} R_{(1)}(v) &= Pr\{x_{(1)} \leq v\} = 1 - Pr\{x_{(1)} > v\}, \\ &= 1 - Pr\{\text{all } x_i > v\} = 1 - [1 - P(v)]^m. \end{aligned} \tag{A.4}$$

The general result, for each rank r:

$$R_{(r)}(v) = Pr\{x_{(r)} \leq v\} = Pr\{\text{at least } r \text{ of the } x_i \leq v\ \}. \qquad (A.5)$$

Or:

$$R_{(r)}(v) = \sum_{i=r}^{m} \binom{m}{i} P^i(v)[1 - P(v)]^{m-i}. \qquad (A.6)$$

In (A.6) it will be very cumbersome, because of the summations, to calculate cumulative probabilities of several ranks of the same sample or to express the CDFs of each rank in analytical functions. Often, they have to be derived for each rank with a fixed sample size. In this case, we can use (A.6) in a recurrent version:

$$R_{(r)}(v) = R_{(r+1)}(v) + \binom{m}{r} P^r(v)[1 - P(v)]^{m-r}, \qquad (A.7)$$

with:

$$R_{(m)}(v) = \binom{m}{m} P^m(v)[1 - P(v)]^{m-m} = P^m(v). \qquad (A.8)$$

When the CDFs of the same rank r have to be determined for different sample sizes it is possible to write (A.6) in an upward recurrent manner, (the superscript ":m" denotes sample size):

$$R_{(r)}^{:m} = R_{(r)}^{:m-1} + \binom{m-1}{r-1} P^r(v)[1 - P(v)]^{m-r}, \qquad (A.9)$$

with, if $m = r$:

$$R_{(r)}^{:r} = R_{(m)}^{:m} = P^m(v). \qquad (A.10)$$

It is also possible to determine the PDFs $r_{(r)}$ of the ranks. Let the variates have PDF $p(v)$. Then, we can use:

$$r_{(r)}(v) = \frac{dP_{(r)}(v)}{dv} = \frac{dP_{(r)}(v)}{dP(v)} \cdot \frac{dP(v)}{dv} = \frac{dP_{(r)}(v)}{dP(v)} \cdot p(v), \qquad (A.11)$$

on (A.6). After differentiating, the following result will be obtained:

$$r_{(r)}(v) = r \binom{m}{r} P^{r-1}(v)[1 - P(v)]^{m-r} p(v). \qquad (A.12)$$

Note, that in the expression of the PDF of the ranks, the CDF of the original variate distribution $P(v)$ is present.

A.2 First and second moments of ranks

In several adaptive OS filters the first moments of the ordered sample are needed. There are various methods to determine the mean of a distribution. The choice depends on which probability function of the variates and the rank is available (CDF or PDF). and whether the calculation

The classical methods are:

$$\begin{aligned} E_v\{x_{(r)}\} &= \int_{v_{min}}^{v_{max}} v\, r_{(r)}(v)\, dv = \int_0^1 v\, dR_{(r)}(v), \qquad (A.13)\\ &= vR_{(r)}(v)\Big|_{v_{min}}^{v_{max}} - \int_{v_{min}}^{v_{max}} R_{(r)}(v)\, dv,\\ E_v\{x_{(r)}\} &= v_{max} - \int_{v_{min}}^{v_{max}} R_{(r)}(v)\, dv, \qquad (A.14)\end{aligned}$$

where $E_v\{\}$ denotes expectation, in this case over v. The lower and upper limit of the domain of v are v_{min} and v_{max}. Depending on whether the CDF or PDF is available a choice is made for the middle expression in (A.13) or the right expression which is extended in (A.14). Note that (A.14) is useful if v has a finite upper bound v_{max}.

Equation (A.13) can be used for calculating the first moment. Using the *Law of the unconscious statistician,* the first moment of a *function* of the rank, $E\{F\{x_{(r)}\}\}$, can also be evaluated [129]:

$$E\{F\{x_{(r)}\}\} = \int_{v_{min}}^{v_{max}} F\{v\} r_{(r)}\, dv. \qquad (A.15)$$

If the integral sum in (A.15) is absolutely convergent, then $E\{g(x_{(r)})\}$ exists. The special cases $g(x_{(r)}) = x_{(r)}^k$, $(x_{(r)} - E\{x_{(r)}\})^k$ and $e^{tx_{(r)}}$ give, respectively, the raw moments, the central moments, and the moment generating function of $x_{(r)}$.

Equation (A.15) is used to find the second moments of the rank orders. These form the diagonal entries of the autocorrelation matrix of ordered samples from a certain distribution:

$$\boldsymbol{C}_{(xx)} = E\left\{ \begin{bmatrix} x_{(1)} \\ x_{(2)} \\ \vdots \\ x_{(m)} \end{bmatrix} \begin{bmatrix} x_{(1)} & x_{(2)} & \dots & x_{(m)} \end{bmatrix} \right\}. \qquad (A.16)$$

With Equation (A.15) the diagonal entries are calculated by solving

$$E\{x_{(r)}x_{(r)}\} = \int_{v_{min}}^{v_{max}} v^2 r_{(r)}(v)\, dv. \qquad (A.17)$$

A.3 Correlation and joint distributions of ranks

The off-diagonal entries in the correlation matrix (A.16) are defined as:

$$E\{x_{(r)}x_{(s)}\} = \int_{w_{min}}^{w_{max}} \int_{v_{min}}^{w} vw \;\, r_{(r)(s)}(v,w) \;\, dv\, dw. \tag{A.18}$$

Here, $r_{(r)(s)}(v,w)$ is the joint PDF of two ordered statistics. A property of the correlation is symmetry:

$$E\{x_{(r)}x_{(s)}\} = E\{x_{(s)}x_{(r)}\}, \tag{A.19}$$

which causes the correlation matrix to be symmetrical. Another property, if the variates originally stem from a distribution symmetric around 0, with odd m is:

$$E\{x_{(r)}x_{(s)}\} = E\{x_{(m-r+1)}x_{(m-s+1)}\}. \tag{A.20}$$

Combining both properties indicates that the correlation matrix is symmetrical in *both* diagonals. In effect, roughly one fourth of the elements have to be calculated, while the others can be copied.

The covariance of the ranks can be found from the correlations and the rank averages with:

$$\begin{aligned} \text{cov.}\{x_{(r)}x_{(s)}\} &= E\{(x_{(r)} - E\{x_{(r)}\})(x_{(s)} - E\{x_{(s)}\})\} \\ &= E\{x_{(r)}x_{(s)}\} - E\{x_{(r)}\}E\{x_{(s)}\}. \end{aligned} \tag{A.21}$$

Which always have the property:

$$\text{cov.}\{x_{(r)}x_{(s)}\} = \text{cov.}\{x_{(s)}x_{(r)}\}. \tag{A.22}$$

If the original distribution is symmetric around its mean value and for odd m, they also share the following property:

$$\text{cov.}\{x_{(r)}x_{(s)}\} = \text{cov.}\{x_{(m-r+1)}x_{(m-s+1)}\}. \tag{A.23}$$

Combining both properties will cause the covariance matrix to be symmetrical in both diagonals.

In order to use (A.18) and (A.21), the joint PDF of two ordered statistics will have to be defined.

Using deduction, we can establish an expression for the joint PDF $r_{(r)(s)}$ of two ranks. If the smaller rank is r and the larger rank is s, then $r-1$ ranks are smaller than r and $m-s$ ranks are larger than s. Using this information, the joint PDF can be constructed:

$$r_{(r)(s)}(v,w) = \frac{m! P^{r-1}(v)p(v)[P(w)-P(v)]^{s-r-1}p(w)[1-P(w)]^{m-s}}{(r-1)!(s-r-1)!(m-s)!}. \tag{A.24}$$

Note that, as in (A.12), the CDF and now also the PDF of the original variates are needed for the expression of the joint PDF of the ordered variates [93].

Because (A.24) is often used if the correlation and the covariances of two ranks has to be calculated, it is efficient to use the following recurrence relations (again $r < s$ so $v \leq w$):

$$r_{(r-1)(s)}(v, w) = \frac{(r-1)[P(w) - P(v)]}{(s-r)P(v)} \cdot r_{(r)(s)}(v, w), \tag{A.25}$$

$$r_{(r-1)(s-1)}(v, w) = \frac{(r-1)[1 - P(w)]}{(m-s+1)P(v)} \cdot r_{(r)(s)}(v, w). \tag{A.26}$$

To be complete, we also derive the joint CDF $R_{(r)(s)}$, using the same reasoning as for the development of (A.6). Suppose there are two samples $x_{(r)}$ and $x_{(s)}$, with $x_{(r)} \leq x_{(s)}$, $r < s$. Then:

$$\begin{aligned} R_{(r)(s)}(v, w) &= Pr\{\text{at least } r\ x_i \leq v, \text{at least } s\ x_i \leq w\}, \\ &= \sum_{j=s}^{m} \sum_{i=r}^{j} Pr\{\text{exactly } i\ x_i \leq v, \text{exactly } j\ x_i \leq w\}, \qquad \text{(A.27)} \\ &= \sum_{j=s}^{m} \sum_{i=r}^{j} \frac{m!}{i!(j-i)!(m-j)!} P^i(v)[P(w) - P(v)]^{j-i}[1 - P(w)]^{m-j}. \end{aligned}$$

For $v > w$, the inequality $x_{(s)} \leq w$ implies $x_{(r)} \leq v$, so that $R_{(r)(s)}(v, w) = R_{(s)}(w)$.

A.4 The uniform and Gaussian distribution

In this section we use the expressions for the statistical properties of the ranks to analyze an ordered sample from a uniform and a Gaussian distribution. First, we will investigate the uniform distribution, because this distribution can be analyzed analytically. We will then show that implicit expressions are not possible for the Gaussian distribution, and therefore analyze it numerically.

Let us first calculate the CDFS of the ranks for a 3-tuple sample drawn from a uniform distribution. The CDF of each variate is:

$$P(v) = v, \qquad (0 \leq v \leq 1). \tag{A.28}$$

Then, for the separate ranks and using the recurrent equation (A.7), the CDFs of the ranks are:

$$R_{(3)}(v) = P^3(v) = v^3,$$

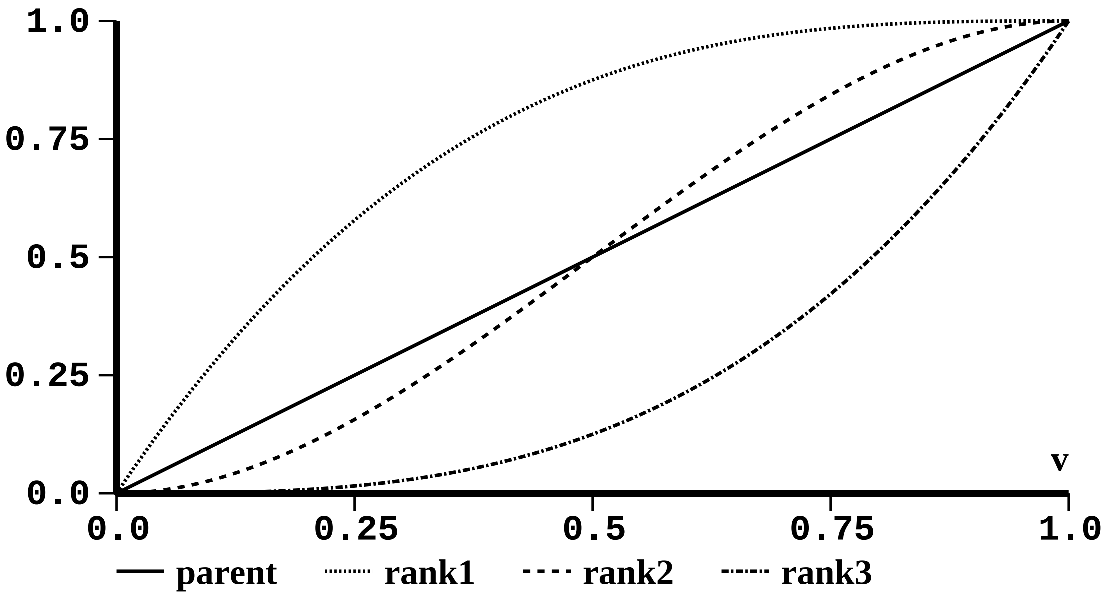


Figure A.1: The CDF curves of the ranks from the uniform distribution

$$\begin{aligned} R_{(2)}(v) &= v^3 + \begin{pmatrix} 3 \\ 2 \end{pmatrix} v^2(1-v) = 3v^2 - 2v^3, \\ R_{(1)}(v) &= 3v^2 - 2v^3 + \begin{pmatrix} 3 \\ 1 \end{pmatrix} v(1-v)^2 = 3v - 3v^2 + v^3. \end{aligned}$$

These three CDFs are plotted in Figure A.1

The expression for the PDF of the ranks is found by using (A.12) with $p(v) = 1$, for $0 \leq v \leq 1$. This results in:

$$\begin{aligned} r_{(3)}(v) &= 3\begin{pmatrix} 3 \\ 3 \end{pmatrix} v^2[1-v]^{3-3} * 1 = 3v^2, \\ r_{(2)}(v) &= 2\begin{pmatrix} 3 \\ 2 \end{pmatrix} v[1-v] * 1 = 6v - 6v^2, \\ r_{(1)}(v) &= \begin{pmatrix} 3 \\ 1 \end{pmatrix} [1-v]^2 * 1 = 3 - 6v + 3v^2. \end{aligned}$$

The PDF curves are plotted in Figure A.2.

To establish the first moments we have used (A.14) which resulted in: $E_v\{x_{(1)}\} = \frac{1}{4}$, $E_v\{x_{(2)}\} = \frac{1}{2}$ and $E_v\{x_{(3)}\} = \frac{3}{4}$. Note that the first rank moments of this symmetrical distribution are symmetrically spaced: $E\{x_{(2)}\} - E\{x_{(1)}\} = E\{x_{(3)}\} - E\{x_{(2)}\}$.

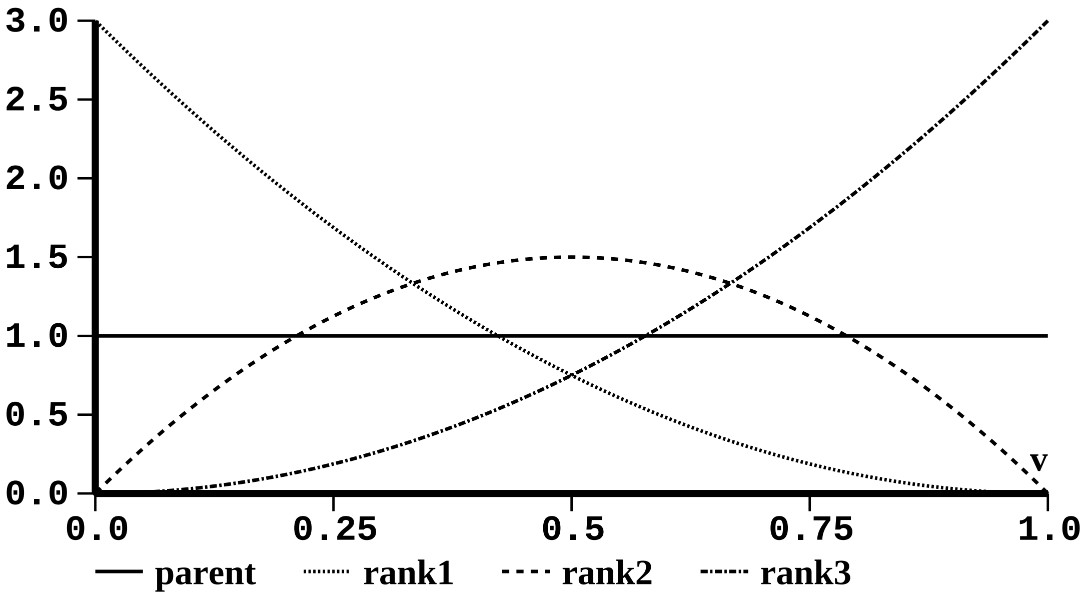


Figure A.2: The PDF curves of the ranks from the uniform distribution

The second moments, the diagonal entries of the correlation matrix (A.16) become:

$$
\begin{aligned}
E\{x_{(1)}x_{(1)}\} &= \int_0^1 v^2(3v^2-6v+3)\,dv = \frac{1}{10}, \\
E\{x_{(2)}x_{(2)}\} &= \int_0^1 v^2(6v-6v^2)\,dv = \frac{3}{10}, \\
E\{x_{(3)}x_{(3)}\} &= \int_0^1 v^2(3v^2)\,dv = \frac{3}{5}.
\end{aligned}
$$

Using (A.18) enables us to establish the correlations.

$$
\begin{aligned}
E\{x_{(2)}x_{(3)}\} &= \int_0^1 \int_0^w vw\,6v\ \ dv\,dw = \frac{2}{5}, \\
E\{x_{(1)}x_{(3)}\} &= \int_0^1 \int_0^w vw\,6(w-v)\ \ dv\,dw = \frac{1}{5}, \\
E\{x_{(1)}x_{(2)}\} &= \int_0^1 \int_0^w vw\,6(1-w)\ \ dv\,dw = \frac{3}{20}.
\end{aligned}
$$

For the uniform distribution, the following joint PDFs for the ranks are found (we have used (A.24), (A.25) and (A.26)):

$$
\begin{aligned}
r_{(2)(3)}(v,w) &= 6v, \\
r_{(1)(3)}(v,w) &= \frac{[w-v]}{v} * r_{(2)(3)}(v,w) = 6w-6v, \\
r_{(1)(2)}(v,w) &= \frac{[1-w]}{v} * r_{(2)(3)}(v,w) = 6-6w.
\end{aligned}
$$

In Table A.1, the results of (A.14) for the first moments, and (A.18) for the corre-

TABLE A.1: FIRST, SECOND MOMENTS AND CORRELATIONS FOR THE UNIFORM $[0,1]$ DISTRIBUTION

r	$E\{x_{(r)}\}$	$E\{x_{(r)}x_{(1)}\}$	$E\{x_{(r)}x_{(2)}\}$	$E\{x_{(r)}x_{(3)}\}$
1	0.250	0.100	0.150	0.200
2	0.500	0.150	0.300	0.400
3	0.750	0.200	0.400	0.600

TABLE A.2: COVARIANCES FOR THE UNIFORM DISTRIBUTION

r	cov.$\{x_{(r)}x_{(1)}\}$	cov.$\{x_{(r)}x_{(2)}\}$	cov.$\{x_{(r)}x_{(3)}\}$
1	0.0375	0.0250	0.0125
2	0.0250	0.0500	0.0250
3	0.0125	0.0250	0.0375

lations are combined. In Table A.2, the results of using (A.21) for the covariances are given.

For a uniform $[-0.5, 0.5]$ distribution, the results in Table A.3 are obtained. The covariances of the $[-0.5, 0.5]$ distribution are of course equal to those of the $[0, 1]$ distribution.

In our first example, we have used the uniform distribution. This is because for this distribution, the rank distributions and moments can be found explicitly, in this case as algebraic equations. However, for the popular Gaussian distribution, the rank distributions and moments cannot be stated explicitly. This is because the CDF of the variates plays a major role in the equations for rank CDF and rank PDF. The integral:

$$P(v > \mu + k\sigma) = Q\{k\} = \frac{1}{\sqrt{2\pi}} \int_k^\infty e^{-\frac{a^2}{2}}\, da, \tag{A.29}$$

which is a variant of the *error function*, cannot be solved in closed form [130]. Here, μ is the mean value and σ is the standard deviation of the distribution.

TABLE A.3: FIRST MOMENTS AND CORRELATIONS FOR THE UNIFORM $[-0.5, 0.5]$ DISTRIBUTION

r	$E\{x_{(r)}\}$	$E\{x_{(r)}x_{(1)}\}$	$E\{x_{(r)}x_{(2)}\}$	$E\{x_{(r)}x_{(3)}\}$
1	-0.250	0.100	0.025	-0.050
2	0.000	0.025	0.050	0.025
3	0.250	-0.050	0.025	0.100

TABLE A.4: FIRST MOMENTS AND CORRELATIONS FOR THE $N\{0,1\}$ DISTRIBUTION

r	$E\{x_{(r)}\}$	$E\{x_{(r)}x_{(1)}\}$	$E\{x_{(r)}x_{(2)}\}$	$E\{x_{(r)}x_{(3)}\}$
1	-0.846	1.276	0.276	-0.551
2	0.000	0.276	0.449	0.276
3	0.846	-0.551	0.276	1.276

TABLE A.5: COVARIANCES FOR THE $N\{0,1\}$ DISTRIBUTION

r	cov.$\{x_{(r)}x_{(1)}\}$	cov.$\{x_{(r)}x_{(2)}\}$	cov.$\{x_{(r)}x_{(3)}\}$
1	0.560	0.276	0.165
2	0.276	0.449	0.276
3	0.165	0.276	0.560

However, the function $Q\{k\}$ is described in extensive tables, by numerical methods. Moreover, for $k > 3$ a quite accurate approximation for $Q\{k\}$ is [131]:

$$\frac{1}{\sqrt{2\pi k^2}}e^{-\frac{k^2}{2}}\left(1-\frac{1}{k^2}\right) < Q\{k\} < \frac{1}{\sqrt{2\pi k^2}}e^{-\frac{k^2}{2}}. \tag{A.30}$$

For our purposes, where we will often work reasonably close to μ, the use of this approximation is very limited. Therefore, the integral will have to be solved by numerical methods. The results are displayed in Table A.4 for the first moments and correlations and in Table A.5 for the covariances of a 3-tuple sample from the standardized Gaussian distribution ($N\{0,1\}$).

Bibliography

[1] M. I. Sezan and R. L. Lagendijk, eds., *Motion analysis and image sequence processing.* Kluwer Academic Publishers, 1993.

[2] H. G. Musmann, P. Pirsch, and H. Grallert, "Advances in picture coding," *Proc. IEEE*, vol. 73, pp. 523–548, Apr. 1985.

[3] V. V. Digalakis, D. G. Manolakis, P. Lazaridis, and V. K. Ingle, "Enhancement of digital angiographic images with misregistration correction and spatially adaptive matched filtering," in *Proc. SPIE Conf. Visual Commun. and Image Processing*, vol. 1818, (Boston, Massachussets, USA), pp. 1264–1270, Nov. 1992.

[4] C. L. Chan, A. K. Katsaggelos, and A. V. Sahakian, "Enhancement of low-dosage cine-angiographic image sequences using a modified expectation maximization algorithm," in *Proc. SPIE Conf. Visual Commun. and Image Processing*, vol. 1660, (Boston, Massachussets, USA), pp. 290–298, Nov. 1992.

[5] G. de Haan, *Motion estimation and compensation.* Ph.d. thesis, Delft University of Technology, Dept. of EE, Delft, the Netherlands, Sept. 1992.

[6] T. S. Huang and Y. P. Hsu, "Image sequence enhancement," in *Image sequence analysis* (T. S. Huang, ed.), ch. 4, pp. 289–309, Springer-Verlag, 1981.

[7] E. Dubois and S. Sabri, "Noise reduction in image sequences using motion compensated temporal filtering.," *IEEE Trans. on Commun.*, vol. COMM-32, pp. 826–831, July 1984.

[8] M. K. Özkan, M. I. Sezan, and A. M. Tekalp, "Adaptive motion-compensated filtering of noisy image sequences," *IEEE Trans. Circuits and Systems for Video Technology*, vol. CSVT-3, pp. 277–290, Aug. 1993.

[9] A. Rosenfeld and A. C. Kak, *Digital picture processing.* New York: Academic Press, 1976.

[10] J. S. Lee, "Digital image enhancement and noise filtering by use of local statistics," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI-2, pp. 165–168, Mar. 1980.

[11] F. C. Jeng and J. W. Woods, "Inhomogeneous Gaussian models for estimation and restoration," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-36, pp. 1305–1312, Aug. 1988.

[12] A. K. Katsaggelos, ed., *Digital image restoration*, vol. 23 of *Springer series in information sciences.* Springer-Verlag, 1991.

[13] R. P. Kleihorst, G. de Haan, R. L. Lagendijk, and J. Biemond, "Motion compensated noise filtering of image sequences," in *Proc. Eusipco '92*, (Brussels, Belgium), pp. 1385–1388, Aug. 24–27, 1992.

[14] R. P. Kleihorst, R. L. Lagendijk, and J. Biemond, "Noise reduction of image sequences using motion compensation and signal decomposition," *IEEE Trans. Image Processing*, vol. IP-4, Mar. 1995.

[15] R. L. Lagendijk and J. Biemond, *Iterative identification and restoration of images.* Boston: Kluwer Academic Publishers, 1991.

[16] R. P. Kleihorst and J. Biemond, "Motion compensated filtering of noisy image sequences," in *Proc. 4th Int. Workshop on Time-Varying Image Processing and Moving Object Recognition*, (Florence, Italy), p. 15, June 10–11 1993.

[17] 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." Submitted to IEEE Proceedings, 1994.

[18] J. N. Driessen, *Motion estimation for digital video.* Ph.d. thesis, Delft University of Technology, Dept. of EE, the Netherlands, Sept. 1992.

[19] R. P. Kleihorst, "Nonlinear filtering of image sequences using order statistics," Chartered Designer's report, Delft University of Technology, Delft, the Netherlands, Jan. 1992.

[20] N. Wiener, *Time series; Extrapolation, interpolation and smoothing of stationary time series.* Cambridge, Massachusetts: The M.I.T. Press, 1949.

[21] R. L. Lagendijk, *Iterative identification and restoration of images.* Ph.d. thesis, Delft University of Technology, Dept. of EE, the Netherlands, Apr. 1990.

[22] M. K. Özkan, M. I. Sezan, and A. T. Erdem, "LMMSE restoration of blurred and noisy image sequences," in *Proc. SPIE Conf. Visual Commun. and Image Processing*, vol. 1606, pp. 743–754, 1991.

[23] A. T. Erdem, M. I. Sezan, and M. K. Özkan, "Motion compensated multiframe Wiener restoration of blurred and noisy image sequences," in *IEEE Proc. Int. Conf. Acoust., Speech, and Signal Proc.*, vol. III, pp. 293–296, 1992.

[24] J. M. Boyce, "Noise reduction of image sequences using adaptive motion compensated frame averaging," in *IEEE Proc. Int. Conf. Acoust., Speech, and Signal Proc.*, (San Fransisco, California, USA), pp. 461–464, 1992.

[25] D. S. Kalivas and A. A. Sawchuck, "Motion compensated enhancement of noisy image sequences," in *IEEE Proc. Int. Conf. Acoust., Speech, and Signal Proc.*, pp. 2121–2124, 1990.

[26] C. B. Dekker, A. J. E. M. Janssen, and P. J. van Otterloo, "The contour plot method for noise reduction in digital video," *Acta Electronica*, vol. 27, no. 1-2, pp. 119–131, 1985.

[27] M. K. Özkan, M. I. Sezan, and A. M. Tekalp, "Motion-adaptive weighted averaging for temporal filtering of noisy image sequences," in *Image Processing Algorithms and Techniques III*, vol. 1657, (San Jose, California, USA), pp. 201–212, Feb. 10–13, 1992.

[28] R. P. Kleihorst, R. L. Lagendijk, and J. Biemond, "Nonlinear filtering of image sequences using order statistics," in *Proc. 12th Symposium on Information Theory in the Benelux*, (Veldhoven, the Netherlands), pp. 49–55, May 23–24 1991.

[29] J. V. Matthews, "Adaptive polynomial filters," *IEEE Signal Processing Magazine*, vol. 8, pp. 10–24, July 1991.

[30] C. L. Chan and B. J. Sullivan, "Nonlinear model-based spatio-temporal filtering of image sequences," in *IEEE Proc. Int. Conf. Acoust., Speech, and Signal Proc.*, vol. 4, (Toronto, Canada), pp. 2989–2992, May 1991.

[31] A. C. Bovik, T. S. Huang, and D. C. Munson, "A generalization of median filtering using linear combinations of order statistics," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-31, pp. 1326–1337, Dec. 1983.

[32] S. S. H. Naqvi, N. C. Gallagher, and E. J. Coyle, "An application of median filters to digital television," in *IEEE Proc. Int. Conf. Acoust., Speech, and Signal Proc.*, (Tokyo, Japan), pp. 2451–2454, 1986.

[33] P. D. Wendt, E. J. Coyle, and N. C. Gallagher, "Stack filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, Aug. 1986.

[34] D. S. Richards, "VLSI median filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-38, pp. 145–153, Jan. 1990.

[35] N. C. Gallagher and G. L. Wise, "A theoretical analysis of the properties of median filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-29, pp. 1136–1141, Dec. 1981.

[36] G. R. Arce, "Multistage order statistic filters for image sequence processing," *IEEE Trans. Signal Processing*, vol. SP-39, pp. 1147–1163, May 1991.

[37] M. B. Alp and Y. Neuvo, "3-dimensional median filters for image sequence processing," in *IEEE Proc. Int. Conf. Acoust., Speech, and Signal Proc.*, vol. 4, (Toronto, Canada), pp. 2917–2920, May 14–17 1991.

[38] A. C. Kokaram and P. J. W. Rayner, "A system for the removal of impulsive noise in image sequences," in *Proc. SPIE Conf. Visual Commun. and Image Processing*, vol. 1818, (Boston, Massachusetts, USA), pp. 322–331, Nov. 1992.

[39] C. T. Lee, B. S. Jeng, R. H. Ju, H. C. Huang, K. S. Kan, J. S. Huang, and T. S. Liu, "Postprocessing of video sequence using motion dependent median filters," in *Proc. SPIE Conf. Visual Commun. and Image Processing*, vol. 1606, (Boston, Massachussets, USA), pp. 728–734, Nov. 1991.

[40] F. L. Lewis, *Optimal control.* New York: Wiley Interscience, 1986.

[41] D. L. Angwin, *Adaptive image restoration using reduced order model based Kalman filters.* Ph.d. thesis, Rensselaer Polytechnic Institute, 1989.

[42] D. Cano and M. Bénard, "3D Kalman filtering of image sequences," in *Image sequence processing and dynamic scene analysis* (T. S. Huang, ed.), vol. F2 of *NATO ASI Series*, pp. 563–579, Springer-Verlag Berlin Heidelberg, 1983.

[43] J. W. Woods and J. Kim, "Motion compensated spatiotemporal Kalman filter," in *Motion analysis and image sequence processing* (R. L. Lagendijk and M. I. Sezan, eds.), ch. 12, Kluwer Academic Publishers, 1993.

[44] R. H. McMann, "Digital noise reducer for encoded NTSC signals," *SMPTE journal*, vol. 87, pp. 129–133, Mar. 1978.

[45] D. I. Crawford, "Spatio/temporal prefiltering for a video conference coder," in *Proc. Int. IEE Conf. on electronic image processing*, (York, U.K.), pp. 236–242, July 1982.

[46] M. I. Sezan, M. K. Özkan, and S. V. Fogel, "Temporally adaptive filtering of noisy image sequences using a robust motion estimation algorithm," in *IEEE Proc. Int. Conf. Acoust., Speech, and Signal Proc.*, vol. 4, (Toronto, Canada), pp. 2429–2432, May 14–17, 1991.

[47] E. Dubois, "Motion compensated filtering of time-varying images," *Multidimensional Systems and Signal Processing*, vol. 3, pp. 211–239, May 1992.

[48] R. P. Kleihorst, R. L. Lagendijk, and J. Biemond, "An order-statistics supported non-linear filter with application to image sequence filtering," in *Proc. IEEE Winter Workshop on Nonlinear Digital Signal Processing*, (Tampere, Finland), pp. 1.2–2.1 to 1.2–2.5, Jan. 17–20, 1993.

[49] T. J. Dennis, "A spatio/temporal filter for television picture noise reduction," in *Proc. Int. IEE Conf. on electronic image processing*, (York, U.K.), pp. 243–249, July 1982.

[50] T. A. Reinen, "Noise reduction in heart movies by motion compensated filtering," in *Proc. SPIE Conf. Visual Commun. and Image Processing*, vol. 1606, pp. 755–763, 1991.

[51] R. C. Triplicane, "Recursive restoration of noisy image sequences," M.Sc. thesis, Northwestern University, Evanston, Il., USA, Oct. 1989.

[52] A. K. Katsaggelos, J. N. Driessen, S. N. Efstratiadis, and R. L. Lagendijk, "Temporal motion-compensated filtering of image sequences," in *Proc. SPIE Conf. Visual Commun. and Image Processing*, vol. 1199, (Boston, Massachusetts, USA), pp. 61–70, 1989.

[53] A. K. Katsaggelos, R. P. Kleihorst, S. N. Efstratiadis, and R. L. Lagendijk, "Adaptive image sequence noise filtering methods," in *Proc. SPIE Conf. Visual Commun. and Image Processing*, vol. 1606, (Boston, Massachusetts, USA), pp. 716–727, Nov. 10–13, 1991.

[54] D. Martinez and J. S. Lim, “Implicit motion compensated noise reduction of motion video scenes,” in *IEEE Proc. Int. Conf. Acoust., Speech, and Signal Proc.*, (Tampa, Florida, USA.), pp. 375–378, 1985.

[55] R. P. Kleihorst, R. L. Lagendijk, and J. Biemond, “Noise reduction of severely corrupted image sequences,” in *IEEE Proc. Int. Conf. Acoust., Speech, and Signal Proc.*, (Minneapolis, Minnesota, USA), April 27–30 1993.

[56] S. T. Alexander, *Adaptive Signal Processing.* New York: Springer Verlag, 1986.

[57] L. Hong and D. Brzakovic, “Bayesian restoration of image sequences using 3-D Markov random fields,” in *IEEE Proc. Int. Conf. Acoust., Speech, and Signal Proc.*, vol. M1.5, pp. 1413–1416, 1989.

[58] S. Geman and D. Geman, “Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images,” *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI-6, pp. 721–741, Nov. 1984.

[59] S. Geman and D. E. McClure, “A nonlinear filter for film restoration and other problems in image processing,” *CVGIP: Graphical models and image processing*, vol. 54, pp. 281–289, July 1992.

[60] J. O. Drewery, R. Storey, and N. E. Tanton, “Video noise reduction,” Technical Report UDC 621.397.238:621.391.83, BBC-RD, July 1984.

[61] J. H. Ahn and J. K. Kim, “Spatio-temporal visibility function for image sequences,” *Electronics Letters*, vol. 27, pp. 585–586, Mar. 1991.

[62] W. Yen, M. J. Wang, and C. Liu, “Performance evaluation of some noise reduction methods,” *CVGIP: Graphical models and image processing*, vol. 54, pp. 134–146, Mar. 1992.

[63] P. C. Cosman, R. M. Gray, and R. A. Olshen, “Evaluating quality of compressed medical images: SNR, subjective rating and diagnostic accuracy,” *Proc. IEEE*, vol. 82, pp. 919–932, June 1994.

[64] R. P. Kleihorst, G. de Haan, R. L. Lagendijk, and J. Biemond, “Noise filtering of image sequences with double compensation for motion,” in *13th Symposium on Information Theory in the Benelux*, (Enschede, the Netherlands), pp. 197–204, June 1–2 1992.

[65] J. K. Aggarwal and N. Nandhakumar, “On the computation of motion from sequences of images,” *Proc. IEEE*, vol. 76, pp. 917–935, Aug. 1988.

[66] S. N. Efstratiadis, *Recursive motion compensated image sequence estimation.* Ph.d. thesis, Northwestern University, Dept. of EE & CS, Evanston, Illinois, USA, Dec. 1991.

[67] B. K. P. Horn and B. G. Schunck, “Determining optical flow,” *Artificial Intelligence*, vol. 17, pp. 185–203, 1981.

[68] R. C. Dorf, ed., *The electrical engineering handbook.* Boca Raton, Florida: CRC Press Inc., 1993.

[69] S. V. Fogel, "Estimation of velocity vector fields from time-varying image sequences," *CVGIP: Image Understanding*, vol. 53, pp. 253–287, May 1991.

[70] A. N. Netravali and J. D. Robbins, "Motion-compensated television coding: Part 1," *The Bell system technical journal*, vol. 58, Mar. 1979.

[71] D. R. Walker and K. R. Rao, "Improved pel-recursive motion compensation," *IEEE Trans. on Commun.*, vol. COM-32, pp. 1128–1134, Nov. 1984.

[72] J. Biemond, L. Looijenga, D. E. Boekee, and R. H. J. M. Plompen, "A pel-recursive Wiener-based displacement estimation algorithm," *Signal Processing*, vol. 13, pp. 399–412, Dec. 1987.

[73] J. C. Brailean and A. K. Katsaggelos, "Recursive displacement estimation and restoration of noisy-blurred image sequences," in *IEEE Proc. Int. Conf. Acoust., Speech, and Signal Proc.*, (Minneapolis, Minnesota, USA), Apr. 1993.

[74] N. M. Namazi and C. H. Lee, "Nonuniform image motion estimation from noisy data," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-38, pp. 364–366, Feb. 1990.

[75] J. M. M. Anderson and G. B. Giannakis, "Noise insensitive motion estimation using cumulants," in *IEEE Proc. Int. Conf. Acoust., Speech, and Signal Proc.*, (Toronto, Canada), pp. 2721–2724, May 1991.

[76] R. P. Kleihorst, R. L. Lagendijk, and J. Biemond, "Motion estimation from noisy image sequences," in *1993 Picture coding symposium*, (Lausanne, Switzerland), March 17–19 1993.

[77] A. W. Lohmann and B. Wirnitzer, "Triple correlations," *Proc. of the IEEE*, vol. 72, pp. 889–901, July 1984.

[78] M. Bierling, "Displacement estimation by hierarchical block matching," in *Proc. SPIE Conf. Visual Commun. and Image Processing*, (Cambridge, Massachusetts, USA), Nov. 9–11, 1988.

[79] G. de Haan and H. Huijgen, "Motion estimator for tv-picture enhancement," in *Proc. 4th Int. Workshop on HDTV and Beyond*, (Torino, Italy), Sept. 3–6, 1991.

[80] G. de Haan, P. W. A. C. Biezen, H. Huijgen, and O. A. Ojo, "True-motion estimation with 3-D recursive search block matching," *IEEE Trans. Circuits and Systems for Video Technology*, vol. VT-3, pp. 368–379, Oct. 1993.

[81] J. M. Mendel, "Tutorial on higher-order statistics (spectra) in signal processing and system theory: Theoretical results and some applications," *Proc. IEEE*, vol. 79, pp. 278–305, Mar. 1991.

[82] J. C. Brailean and A. K. Katsaggelos, "Simultaneous recursive displacement estimation and restoration of image sequences," in *Proc. IEEE Conf. on Sys., Man, & Cyber.*, (Chicago, Illinois, USA), pp. 1574–1579, Nov. 1992.

[83] A. K. Katsaggelos and K. T. Lay, *Identification and restoration of images using the expectation maximization algorithm.* Springer-Verlag, 1991.

[84] P. J. Brockwell and R. A. Davis, *Time series: theory and methods.* New York: Springer Verlag, 1986.

[85] A. D. Hillery and R. T. Chin, "Restoration of images with nonstationary mean and autocorrelation," in *IEEE Proc. Int. Conf. Acoust., Speech, and Signal Proc.*, (New York, USA), pp. 1008–1011, 1988.

[86] S. Y. Park and Y. H. Lee, "Double smoothing of images using median and Wiener filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-37, pp. 943–946, June 1989.

[87] G. Strang, *Linear algebra and its applications, 2nd ed.* New York: Academic Press, 1980.

[88] N. C. Callagher and G. L. Wise, "A theoretical analysis of the properties of median filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-29, pp. 1136–1141, Dec. 1981.

[89] A. Restrepo and A. C. Bovik, "Adaptive trimmed mean filters for image restoration," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-36, pp. 1326–1337, Aug. 1988.

[90] I. Pitas and A. N. Venetsanopoulos, *Nonlinear digital filters: principles and applications.* Kluwer Academic Press, 1990.

[91] I. Pitas and A. N. Venetsanopoulos, "Order statistics in digital image processing," *Proc. IEEE*, vol. 80, pp. 1892–1921, Dec. 1992.

[92] E. H. Lloyd, "Least squares estimation of location and scale parameters using order statistics," *Biometrika*, vol. 39, pp. 88–95, 1952.

[93] H. A. David, *Order statistics, 2nd ed.* New York: Wiley & Sons, 1981.

[94] K. J. Astrom and B. Wittenmark, *Adaptive Control.* New York: Addison-Wesley, 1989.

[95] E. Eleftheriou and D. D. Falconer, "Steady state behavior of RLS adaptive algorithms," in *IEEE Proc. Int. Conf. Acoust., Speech, and Signal Proc.*, (Tampa, Florida, USA), March 1985.

[96] A. E. Sarhan and B. G. Greenberg, eds., *Contributions to order statistics.* New York: John Wiley & Sons, 1962.

[97] P. J. Rousseeuw and A. M. Leroy, *Robust regression and outlier detection.* New York: Wiley & Sons, 1987.

[98] R. V. Hogg, "An introduction to robust estimation," in *Robustness in Statistics* (R. L. Launer and G. N. Wilkinson, eds.), pp. 1–17, Academic Press, 1979.

[99] H. Stark and J. W. Woods, *Probability, random processes and estimation theory for engineers.* Englewood Cliffs, New Jersey: Prentice Hall, 1986.

[100] R. Bernstein, "Signal adaptive two-dimensional noise filtering using local signal features," in *SIGNAL PROCESSING II: Theories and Applications* (H. W. Schüssler, ed.), pp. 239–242, Elsevier Science Publishers B.V. (North-Holland), 1983.

[101] D. T. Kuan, A. A. Sawchuck, T. C. Strand, and P. Chavel, "Adaptive noise smoothing filter for images with signal-dependent noise," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI-7, pp. 165–177, Mar. 1985.

[102] C. L. Chan, B. J. Sullivan, A. V. Sahakian, A. K. Katsaggelos, T. Frohlich, and E. Byrom, "Spatio-temporal filtering of angiographic image sequences corrupted by quantum mottle," in *Proc. SPIE Conf. Visual Commun. and Image Processing*, vol. 1450, (Boston, Massachussets, USA), pp. 208–217, Nov. 1991.

[103] C. L. Chan, A. K. Katsaggelos, and A. V. Sahakian, "Image sequence filtering in quantum-limited noise with applications to low-dose fluoroscopy," *IEEE Trans. Medical Imaging*, vol. MI-12, Sept. 1993.

[104] M. Rabbani, "Bayesian filtering of Poisson noise using local statistics," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-36, pp. 933–937, June 1988.

[105] M. Basseville, "Edge detection using sequential methods for change in level- part 2: sequential detection for change in mean," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-29, pp. 32–48, Feb. 1981.

[106] W. S. Agee and R. H. Turner, "Application of robust regression to trajectory data reduction," in *Robustness in statistics* (R. L. Launer and G. N. Wilkinson, eds.), pp. 107–126, Academic Press, 1979.

[107] F. R. Hampel, "The influence curve and its role in robust estimation," *J. Am. Statist. Assoc.*, vol. 69, pp. 383–393, June 1974.

[108] R. L. Launer and G. N. Wilkinson, *Robustness in Statistics.* New York: Academic Press, 1979.

[109] P. J. Huber, "Robust estimation of a location parameter," *Ann. Math. Statist.*, vol. 35, pp. 73–101, 1964.

[110] P. W. Holland and R. E. Welsh, "Robust regression using iteratively reweighted least squares," *Communication Statistics*, vol. A6, pp. 813–828, 1977.

[111] V. Barnett and T. Lewis, *Outliers in statistical data, 2nd ed.* Chisester: John Wiley & Sons, 1983.

[112] J. van Soest, *Elementaire statistiek.* Delft, the Netherlands: Delftse uitgevers maatschappij, 1985.

[113] H. L. Harter, D. S. Clemm, and E. H. Guthrie, "The probability integral and percentage points of the studentized range; critical values for Duncan's new multiple range test," Technical Report 58-484, vol 2., WADC, 1959.

[114] D. B. Duncan, "Multiple range and multiple *f*-tests," *Biometrics*, vol. 11, pp. 1–42, 1955.

[115] M. L. Giger, K. Doi, and H. Fujita, “Noise Wiener spectra of II-TV digital imaging systems,” *Med. Phys.*, vol. 13, pp. 131–138, Mar. 1986.

[116] R. P. Kleihorst, R. L. Lagendijk, and J. Biemond, “Order statistics supported filtering of low-dosage X-ray image sequences,” in *8th Workshop on Image and Multi-Dimensional Signal Processing*, (Cannes, France), pp. 94–95, September 8–10 1993.

[117] S. I. Resnick, *Adventures in stochastic processes.* Boston: Birkhäuser, 1992.

[118] F. A. Haight, *Handbook of the Poisson distibution.* New York: John Wiley & Sons, 1966.

[119] N. Balakrishnan and A. C. Cohen, *Order Statistics and inference; estimation methods.* San Diego: Academic Press, 1991.

[120] H. J. Malik, N. Balakrishnan, and S. E. Ahmed, “Recurrence relations and identities for moments of order statistics, 1: arbitrary continuous distribution,” *Commun. Statist.-Theory Meth.*, vol. 17, no. 8, pp. 2623–2655, 1988.

[121] R. P. Kleihorst, R. L. Lagendijk, and J. Biemond, “A robust l-estimator for filtering quantum-limited image sequences,” in *Proc. Eusipco '94*, (Edinburgh, Scotland), Sept. 13-16, 1994.

[122] A. van der Ziel, *Noise.* New York: Prentice Hall inc., 1954.

[123] G. Cortelazzo, G. A. Mian, and R. Parolari, “Statistical characteristics of granular camera noise.” To appear in IEEE Transactions on Circuits and Systems for Video Technology, 1994.

[124] R. P. Kleihorst, R. L. Lagendijk, and J. Biemond, “An efficient spatio-temporal os-filter for gamma-corrected video signals,” in *Proc. IEEE Int. Conf. on Image Processing*, (Austin, Texas), Nov. 13-16, 1994.

[125] S. Haykin, *Adaptive Filter theory, 2nd ed.* Englewood Cliffs: Prentice Hall, 1991.

[126] C. M. Lo and A. A. Sawchuck, “Nonlinear restoration of filtered images with Poisson noise,” in *Proc. SPIE Conf. Visual Commun. and Image Processing*, vol. 207, pp. 84–95, 1979.

[127] S. P. Kim, N. K. Bose, and H. M. Valenzuela, “Recursive reconstruction of high resolution image from noisy undersa mpled frames,” *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-38, pp. 1013–1027, June 1990.

[128] B. M. Sadler and G. B. Giannakis, “Shift- and rotation-invariant object reconstruction using the bispec trum,” *Journal of the Optical Society of America: A*, vol. 9, pp. 57–69, Jan. 1992.

[129] G. Grimmet and D. Stirzaker, *Probability and random processes.* Oxford: Clarendon Press, 1982.

[130] A. B. Carlson, *Communication systems.* Singapore: McGraw-Hill, 1986.

[131] J. M. Wozencraft and I. M. Jacobs, *Principles of communication engineering.* New York: Wiley, 1965.

Samenvatting

Ruisfilteren van beeldsequenties

Onder beeldsequenties verstaan we gedigitaliseerde opnamen van 3 dimensionale dynamische scenes. Het komt nogal eens voor dat een beeldsequentie verruist is. Deze ruis kan bijvoorbeeld zijn onstaan in de beeldsensor of in andere gedeeltes van het beeldvormende systeem. Het is aan te bevelen om deze ruis te reduceren als er sprake is van verdere verwerking van de beeldsequentie of natuurlijk ten behoeve van de visuele waardering. De reductie van de ruis is de taak van het ruisfilter: deze moet de ruis zoveel mogelijk onderdrukken zonder te veel vervorming van de originele beeldinformatie. Het “ruisfilteren van beeldsequenties” is het onderwerp van dit proefschrift.

Het beeldsequentiesignaal heeft drie coördinaten: Twee om de positie in het beeldvlak aan te duiden voor de spatiële informatie, de beelden. Eén in de tijd om de temporele richting te indexeren, het beeldnummer. De signalen in deze richtingen, de “spatiële” en “temporele” signalen, hebben verschillende eigenschappen. Voor ruisfilteren is het belangrijk om deze eigenschappen te kennen. Ze hebben betrekking op de homogeniteit en voorspelbaarheid van het signaal. De spatiele signalen beschrijven de beeldinformatie en zijn daarom meestal niet stationair, waardoor ze moeilijk zijn te filteren zonder de beeldinformatie te vervormen. Het temporele signaal kunnen we in twee vormen tegenkomen: het kan van een niet actief gedeelte van de beeldsequentie, zoals een stilstaande achtergrond komen waar het een homogeen karakter heeft en goed te filteren valt; of het kan komen van een actief gedeelte van de beeldsequentie waarin zich bewegende objecten voordoen. In dit laatste geval is het temporele signaal ten hoogste homogeen binnen kleine tijdsvensters maar over het geheel niet-stationair en daardoor moeilijk te verwerken.

De oorzaken van de niet-stationaire gedeeltes in de temporele signalen zijn beweging, scenewisseling, of verandering van belichtingscondities. Een voor de hand liggende oplossing om de niet-stationaire situaties grotendeels te vermijden is om het temporele signaal voor beweging te compenseren. Dit komt neer op niet meer langs de tijdas, maar langs het traject van de beweging te filteren. Op deze manier hebben we te maken met “bewegings-gecompenseerd” filteren. Vanwege het meer homogene, temporele signaal zijn de uiteindelijke resultaten beter. Meestal zijn de schattingen van de beweging echter vrij onnauwkeurig vanwege de ruis en de onvolledigheid van het bewegingsmodel dat bij het bewegingsschatten gehanteerd wordt. Hierdoor, en doordat scene- en belichtingswisselingen ook niet-stationaire situaties veroorzaken, zullen er zich nog steeds niet-stationairiteiten voordoen in het temporele signaal.

Om de problemen veroorzaakt door de ruis bij het bewegingsschatten enigszins te kunnen opvangen wordt in Hoofdstuk 3 een ruisrobuuste bewegingsschattingsmethode beschreven. Deze methode is een gemodificeerde recursieve block-matcher en is in originele vorm bekend vanwege zijn consistente resultaten en tamelijke ongevoeligheid voor observatieruis. Om de ruisrobuustheid te vergroten hebben we enkele modificaties toegepast. Hieronder valt een nieuw schattingscriterium dat gebaseerd is op derde orde statistieken en daardoor transparant is voor symmetrische ruis.

Omdat de signalen, zelfs in bewegingsgecompenseerde beeldsequenties, meestal temporeel noch spatiëel stationair zijn, is het vereist om gebruik te maken van adaptieve filtertechnieken. Adaptieve filters hebben de mogelijkheid om zich aan te passen aan de globale of lokale kenmerken van het signaal. Het succes van lokaal adapterende filtertechnieken wordt duidelijk uit het exposé in de hoofdstukken 2 en 3 van dit proefschrift waar verschillende bekende filtermethodes de revue passeren. In een experimentele vergelijking blijkt dat alleen lokaal adapterende filters succes hebben in het vervormingsarm ruisfilteren van praktische beeldsequenties. Uit onze experimenten blijkt bovendien dat voor een goed resultaat bewegingscompensatie bij deze filters niet altijd nodig is.

Een klassieke methode om niet-stationaire signalen te filteren is om gebruik te maken van trend-verwijdering en normalisatie. Na verwijderen van de trend, het locale gemiddelde, en na normalisatie met de locale deviatie, blijft een homogeen signaal over dat vrijwel alle ruis bevat en bovendien relatief eenvoudig te filteren is. In feite bevatten de trend en de normalisatie-coëfficiënt de niet-stationairiteiten. Na het ruisfilteren wordt de normalisatie ongedaan gemaakt en de trend weer toegevoegd om het uiteindelijke resultaat te verkrijgen. Deze klassieke methode is op grote schaal toegepast in de tijdreeksverwerking maar slechts zelden bij beeldfilteren. In Hoofdstuk 3 van dit proefschrift hebben we deze techniek, de “decompositiemethode” genoemd, weten toe te passen op het probleem van beeldsequentie ruisfilteren.

Zoals eerder gezegd was, komt de trend overeen met het lokale gemiddelde en de normalisatiefactor met de lokale deviatie. Speciaal voor de decompositiemethode hebben we een schatter voor deze lokale statistieken ontwikkeld. De schatter bepaalt het lokale gemiddelde en de lokale deviatie met behulp van in waarde geordende observaties binnen een spatio-temporeel venster. Een kenmerk van de nieuwe schatter is dat hij optimaal is voor verschillende practische vormen van ruis. Bovendien adapteert de schatter via metingen in de observatie naar de verdeling van de observatieruis.

Een bruikbare en populaire filterstructuur is te vinden in het LLMMSE-filter waarbij de afkorting staat voor de eigenschappen van het filter: een lineaire expressie die lokaal optimale resultaten geeft. Parameters van het filter zijn schattingen van de locale statistieken, het lokale gemiddelde en de lokale variantie. Voor scherpe en bruikbare resultaten zijn nauwkeurige schattingen van de lokale statistieken nodig. In de meeste gepubliceerde methodes wordt als schatter een eenvoudig algoritme gebruikt dat bij niet-stationaire situaties in het signaal sub-optimale resultaten oplevert. In Hoofdstuk 5 behandelen we het gebruik van de schatter uit Hoofdstuk 4, die schat op geordende observaties, ter ondersteuning van het LLMMSE-filter. Bovendien bespreken we twee methodes om een homogeen gedeelte uit het spatio-temporele schattings-venster te selecteren dat voor een goede schatting borg staat. We demonstreren een iteratieve selectiemethode die binnen de schattingsmethode te gebruiken valt en een efficiënte selectiemethode op basis van een statistische test van de geordende observatiewaardes. De nieuwe combinatie van selectie, schatter en LLMMSE-filter geeft bijzonder goede filterresultaten.

Tot dusver hebben we in dit proefschrift alleen nog maar gekeken naar onafhankelijke observatieruis. Er zijn practische situaties denkbaar waar signaalafhankelijke ruis voorkomt. Hoofdstuk 6 behandelt twee van deze situaties: Röntgensequenties en gamma gecorrigeerde beeldsequenties.

Binnen de medische wereld wordt de Röntgendosis bij het maken van medische beeldsequenties zo klein mogelijk gehouden. Bovendien wordt de belichtingstijd per beeld kort gehouden om bewegingsvervorming te vermijden. Het resultaat is dat slechts een beperkt aantal Röntgendeeltjes de beeldinformatie moet genereren. Hierdoor is er inherent met deze beeldvormingsmethode een vorm van ruis geïntroduceerd die signaalafhankelijk is. Speciaal voor het schatten van de diagnostische informatie uit de verruiste beeldsequentie ontwikkelen we in het eerste gedeelte van Hoofdstuk 6 een filtermethode. Deze methode maakt weer gebruik van geordende observaties. Zij wordt ondersteund door de iteratieve selectiemethode uit Hoofdstuk 5 om homogene gedeeltes uit het spatio-temporele schattingsvenster te selecteren.

Om beeldsequenties natuurgetrouw weer te geven op een televisiescherm wordt er gammacorrectie, een niet-lineaire bewerking, toegepast. Een neveneffect van deze bewerking is dat ruis, aanwezig voor de bewerking, nu signaalafhankelijk is geworden. Om deze ruis te reduceren wordt in het tweede gedeelte van Hoofdstuk 6 een filtermethode ontwikkeld bestaande uit de selectiemethode op basis van een statistische test, afgeleid in Hoofdstuk 5, en een verzameling van schatters waaruit de juiste geselecteerd wordt. Omdat de schatters constante, optimale coëfficiënten hebben, is de gehele methode erg efficiënt.

Als conclusie van dit promotieonderzoek kan gesteld worden dat voor een bruikbaar filterresultaat een locaal adapterend filter noodzakelijk is. Bij het gebruik van sterk adapterende filters blijkt dat het toevoegen van bewegingscompensatie slechts tot detailverbetering leidt. Dit proefschrift introduceert het gebruik van "order-statistic" filters met locaal adapterende gewichten in het kader van het ruisfilteren van beeldsequenties. Uit een vergelijking met andere methodes blijkt dat de geintroduceerde methodes zeer sterke kandidaten zijn. De prestaties kunnen bovendien nog opgevoerd worden door met behulp van robuste technieken homogene signaalgedeeltes te selecteren. Observatievergelijkingen met signaalafhankelijke ruis kunnen eenvoudig in een vorm gebracht worden die door de voorgestelde methodes gefilterd kunnen worden.

Acknowledgements

This thesis reflects the work I carried out as a Ph.D. student in the Information Theory Group at Delft University of Technology, the Netherlands. Although the actual Ph.D. program lasted for only a couple of years, I have been dwelling in this group for a considerable larger number of years.

Now that most of the work seems to have converged, acknowledging some persons is in place. I would like to thank my Professor Jan Biemond and my supervisor Inald Lagendijk for the stimulating discussions, for their educative support, and for accepting my desire for freedom.

A number of persons from outside the group have been very helpful by sharing their experience, time and enthusiasm. Among those are Professor Aggelos Katsaggelos, Makis Efstratiadis and Jim Brailean while affiliated with Northwestern University. Thanks to funding by NATO under grant 0103/88, we were able to do some joint research. Thanks are also due to Cees Slump from Twente University and last but not least, to Cock Storm, Gerard de Haan, Hans Driessen and Richard Jonkman.

Also, I want to thank everyone connected with the Information Theory Group for sharing some hard, barely livable but positively remembered years. I thank my family, friends and especially Annemieke for supporting those years emotionally (and financially) and my classic car for not stranding. The road hasn't been easy, but it was worth all the miles.

Richard P. Kleihorst, August 1994.

Curriculum vitae

Richard P. Kleihorst was born in Schiedam, the Netherlands, on June 22^{nd}, 1965. From 1982 to 1983 he attended the Rythovius College in Eersel, the Netherlands, to acquire his V.W.O. diploma. He received the M.Sc. degree in Electrical Engineering and the Chartered Designer's certificate in Information Technology from Delft University of Technology, the Netherlands, in 1989 and 1992, respectively. From February 1992 until June 1994 he worked towards the Ph.D. degree in the Information Theory Group at Delft University of Technology. In June 1994 he joined the Philips Research Laboratories, Eindhoven, the Netherlands. His interests include: restoration of images, image sequences, classic cars, bikes, radio receivers, televisions and a Belgian house.