Quality of noise reduction in TV-systems

Puttenstein, J.G.

Award date:
2001

TU/e technische universiteit eindhoven

Faculty of Electrical Engineering
Section Design Technology For Electronic Systems (ICS/ES)
ICS-ES 759

Master Thesis

QUALITY OF NOISE REDUCTION IN TV-SYSTEMS

Ir. J.G. Puttenstein

Coach:	Dr. I.E.J. Heynderickx (Philips Research Laboratories)
Supervisor:	Prof. Dr. Ir. G. de Haan
Date:	March 2001

Quality of noise reduction in TV-systems

Master Thesis

Graduation committee:
Prof. Dr. Ir. G. de Haan,
Dr. I.E.J. Heynderickx,
Dr. Ir. J.P.M. Voeten,
Eindhoven University of Technology,
March 2001.

The work described in this report has been carried out
at the PHILIPS RESEARCH LABORATORIES Eindhoven, the Netherlands,
as part of the Electrical Engineering study programme of the Eindhoven University of Technology.

Quality of noise reduction in TV-systems
Master Thesis, Eindhoven University of Technology.
-List of references and
-Summary in English, included.
Keywords: noise reduction, quality measures, filters, TV-systems.

Acknowledgments

I would like to thank Gerard de Haan for giving me the opportunity to do my graduation project at the Philips Research Laboratories. I am honoured and grateful that I have been able to work with him. The same is true for my other supervisor, Ingrid Heynderickx. Her helpful criticism, in the preparation of the perceptual experiments and also during the writing of this report were of much value to me.

Thank you Jeroen Voeten for reading my report in such a short period of time.

I thank all the colleagues of the Video Processing and Visual Perception Group and especially Geert de Povere for making me a part of their group. I liked working with them very much.

Last and not least, I would like to thank my mother for her support during the whole of my study, without which I would not have been able to graduate so successfully.

Summary

In the processing of video information, noise enters the system in almost every part of the path of the video signal from the source, e.g. the video camera, to the viewing system, e.g. television or computer screen. The two most important kinds of noise are Gaussian noise and impulse noise. Gaussian noise that has a limited bandwidth is also called pink noise. Noise can be added to the video signal (called additive noise) or it can be multiplied with the video signal (multiplicative noise). In this report only additive Gaussian bandwidth limited white noise will be considered. The aim of this report is to find noise reduction algorithms that are suitable for TV-systems, evaluate the quality of these algorithms with existing evaluation measures and, hopefully, find a better quality measure to describe the subjective performance of noise reduction algorithms.

Noise reduction algorithms

Noise reduction algorithms can be described by many characteristics, such as filter support, support size, filter structure types, noise filter features, neighbourhood selection and adaptivity, statistics used in filtering and complexity and hardware constraints. Not all noise reduction algorithms are suitable for use in TV-systems. A selection of available filters is described and implemented.

Evaluation measures

Subsequently, noise reduction algorithm evaluation measures have been found. There are basically three types: objective, hybrid and subjective evaluation measures. While objective measures are measures that give a relation between luminance properties of video sequences, hybrid measures try to incorporate human vision characteristics. The objective evaluation measures used are: *SNR*, *PSNR*, *SNRI/PSNRI*, *MSE*, *MD*, textitMB, *ST*, *CPR*, *FFNSA* and *DPC*. The hybrid evaluation measures are: *SDM*, *TDM*, *Mdis*, Fujio *SNR* and *PSNR*, CCIR 4211 *SNR* and *PSNR* and CCIR 5672 *SNR* and *PSNR*.
Subjective measures, also called subjective panel tests, gather opinions from observers directly.
The objective and hybrid evaluation measures have been used to evaluate the implemented noise reduction algorithms. Also two subjective panel tests have been conducted in each of which five noise reduction algorithms were compared among each other on quality perception and remaining noise in the filtered video. Also information on the important parts that observers used to judge

quality and noise was gathered.

Objective evaluation results

The noise algorithms are ranked on the overall performance for 26 and 32 dB *PSNR* sequences. The ranking of the algorithms is for the 26 dB *PSNR* sequences, in order of goodness: SWAN_DNR, Siemens, DNR_mc, SWAN, Median_Fuzzy, DNR and Falcon_mc_hp. For 32 dB *PSNR* sequences, the ranking, in order of goodness, is: SWAN_DNR, SWAN, DNR_mc, Median_Fuzzy, DNR, Siemens and Falcon_mc_hp. From the results, it became clear that there was a strong correlation among the *SNR*, *PSNR*, *SNRI*, *MD*, *MSE* and *MB* evaluation measures for 26 dB *PSNR* sequences. For 32 dB *PSNR* sequences, the same group of evaluation measures showed a correlation, only small differences arose for the *MSE* evaluation measure, while the ranking of the noise reduction algorithms with *MB* deviated heavily from the previously mentioned ones.
Another group of evaluation measures that showed a highly consistent behaviour is the group of the weighted *SNR* and *PSNR* evaluation measures. There were only small deviations within this group.

Subjective evaluation results

The conducted subjective test containing two experiments gave results on quality and noise performance of the noise reduction algorithms and on important parts that observers use to evaluate quality and noise performance.

Overall in experiment 1, the Siemens algorithm was the best, for both quality as noise performance. Next came SWAN_DNR together with Falcon_mc_hp. SWAN could not be distinguished on noise performance from Falcon_mc_hp. On quality performance, SWAN was worse than SWAN_DNR and Falcon_mc_hp. The worst performing algorithm for both quality and noise performance was the Median_Fuzzy algorithm.

Subsequently, the effect of noise strength on quality and noise performance has been described. This is done by examining the quality and noise performance for 26 dB and for 32 dB *PSNR* sequences separately.
For 26 dB *PSNR* sequences, the Siemens algorithm performed the best, followed by the SWAN_DNR algorithm, then came Falcon_mc_hp and SWAN that were comparably good. The worst algorithm was Median_Fuzzy.
For 32 dB *PSNR* sequences, the Siemens noise reduction algorithm was the best perceived algorithm, next came the Falcon_mc_hp and the SWAN_DNR algorithms and, finally, the SWAN_DNR and Median_Fuzzy algorithms. The SWAN_DNR algorithm had a combined second and third rank, together with, respectively, Falcon_mc_hp and Median_Fuzzy. Results for quality and noise performance for the two different noise strengths were exactly the same.

In experiment 2, the SWAN_DNR, DNR_mc and Falcon_mc_hp algorithms scored the best. These algorithms were undistinguishable on basis of quality perfor-

mance. Next came the SWAN algorithm, followed by the DNR noise reduction algorithm that performed the worst on quality. The noise performance was the same, only Falcon_mc_hp and SWAN algorithms could not be distinguished from each other. For both noise and quality performance of noise reduction on 26 dB *PSNR* sequences, SWAN_DNR scored the best (on noise SWAN_DNR was comparably good as DNR_mc). DNR performed the worst for both scores. DNR_mc and Falcon_mc_hp scored comparably for both quality and noise performance and were better performing than DNR. For both quality and noise performance for 32 dB *PSNR* sequences, DNR_mc, SWAN_DNR, Falcon_mc_hp and SWAN scored better than the DNR noise reduction algorithm. DNR scored the worst for quality and noise.

The observers were also asked to indicate important areas for judgement of quality and noise. Results showed that for the sequences *Renata* and *Car & Gate* no explicit preference can be given for detailed or undetailed parts. For the *Teeny* sequence a clear preference is present for undetailed areas of the sequence. These results are applicable for quality as well as noise judgement.

Regression model of subjective quality with objective and subjective evaluation measures

The scores for quality and noise performance obtained from the two subjective experiments have been averaged over the three sequences. It appeared from a correlation analysis with SPSS that the averaged quality and noise scores are highly correlated. The correlation coefficients of all objective and hybrid evaluation measures with the quality performance have been calculated. The results show that correlation in general of the objective and hybrid evaluation measures with quality as measured in the two subjective experiments is very low. The best correlating independent measures are *CPR* and *MB*. These evaluation measures have been used for the two regression models of the results of experiment separately, and for the regression model of the combined results of the two experiments. The regression model of the first experiment gives a very low explanatory power of 50% (R^2 square value of 0.503) of the variance of the subjective quality. The regression model of the second experiment gives a satisfactory explanatory power of 84% (R^2 square value of 0.837). The regression model for the combined results of experiment 1 and 2 gives a low explanatory power of 60% (R^2 square value of 0.593). The coefficents of the three regression models are close together, which means that the model is consistent for the individual experiments and the combined experimental results. The explanatory power of the model for the combined experimental results is with 60% very low.

Contents

Glossary

Definitions

image	frame or field
video	succession of interlaced frames
LF	low-frequency components
HF	high-frequency components

Symbol	Description
$F(\vec{x}, n)$	original (noise-free) video sequence or image at spatial position $\vec{x} = (x, y)$ and temporal position n
$G(\vec{x}, n)$	noise impaired video sequence or image at spatial position $\vec{x}$ and temporal position n
$G_F(\vec{x}, n)$	noise reduced version of G at spatial position $\vec{x}$ and temporal position n
n	frame or field number

List of Tables

List of Figures

Chapter 1

Introduction

In the processing of video information, noise enters the system in almost every part of the path of the video signal from the source, e.g. the video camera, to the viewing system, e.g. television or computer screen. The two most important kinds of noise are Gaussian noise and impulse noise. Gaussian noise that has a limited bandwidth is also called pink noise. Noise can be added to the video signal (called additive noise) or it can be multiplied with the video signal (multiplicative noise). In Figure 1.1 and Figure 1.2 a video image is shown with and without pink[1] noise of different strengths, resulting in images with a *PSNR* of 20 dB and a *PSNR* of 26 dB.

Figure 1.1: *Teeny with a PSNR of 20 dB and noiseless Teeny*

[1] By pink noise is meant here: Gaussian white noise, filtered with a low-pass filter of 5MHz.

Figure 1.2: *Teeny with a PSNR of 26 dB and noiseless Teeny*

1.1 Sources of noise

Almost in the whole path of the video signal, from the recording to the display on the consumer TV-system, noise is added. The most important parts of the video chain where noise is added, are the transmission path of the PAL-encoded video signal and the reception of the video in the consumer TV-system.

1.2 Noise reduction

In addition to a very careful treatment of the video signal to *avoid* noise, filters can be used to *reduce* the negative effect of noise on the quality of video. The working principle of the noise filter is called the noise reduction algorithm[2]. Filters are used among others inside the consumer application to reduce the noise before the video appears to the user. The noise filter can enhance the video quality considerably and is, therefore, an important feature of a consumer application.
The performance of a noise filter can be measured by applying noise reduction quality measures. The purpose of a quality measure is twofold. Firstly, the measure should facilitate an accurate evaluation of the quality of a noise reduction algorithm. Secondly, it should be commonly used. Basically, two approaches to assess noise reduction quality are in use:

- Objective measures. These measures describe the objective qualifications, such as noise strength and (objective) differences between noise reduced and noise-free images.
- Subjective measures. Quality of noise-reduced video as perceived by humans. Quality is, in the case of relative small degradations of the video, mostly described as a relative measure with respect to the noise-less video.

[2]In this report, no distinction is made between the terms 'noise filter' and 'noise reduction algorithm'.

Or, as used in this report (Chapter 5), quality can be described by comparing noise reduced video sequences, resulting from different noise reduction algorithms, among each other.

1.3 Objective

The objective of this report is threefold:

1. Find noise reduction algorithms that are suitable for consumer applications, given their complexity, architecture and performance.
2. Evaluate the noise reduction algorithms with the help of a subjective quality experiment and try to correlate the results with objective quality measures.
3. Recommend a video noise reduction quality measure to improve the objective evaluation of video noise reduction algorithms.

1.4 Constraints

There are certain limitations that should be posed upon the subject of video noise reduction to narrow the research topic to realistic proportions. Of course, these constraints should be in accordance with the objectives of the report. The limitations will limit the scope of this report further to noise reduction in standard definition consumer systems. This constraint has the following consequences:

- The noise is bandwidth limited from 0 to 5MHz.
- The noise is additive Gaussian white noise.
- The noise source is the transmission phase of the broadcasting path. This means that the video signal is PAL encoded when noise is added.
- Algorithms are only considered if they are applicable in television systems. This demand limits the maximum complexity and memory usage of the algorithms that can be used.
- Only luminance filtering is taken into account.

The implemented algorithms are not real-time. They are tested in a simulation environment. The focus of the implementation was on the correct processing, rather than on efficient memory and CPU usage.

1.5 Organisation of this report

Chapter 2 gives a classification of noise reduction algorithms by examining the different characteristics of the algorithms (objective 1). In addition, it gives elaborate descriptions of seven different noise reduction algorithms. Chapter 3

looks into ways to evaluate the quality of noise reduction algorithms. Objective, perceptual and hybrid quality measures are introduced. Chapter 4 gives the experimental set-up of the two subjective experiments that were conducted to evaluate the perceived quality of seven noise reduction algorithms. All the noise reductions used in the subjective experiments are described in the last part of Chapter 2[3]. In addition to that, the results of the objective measures and of the subjective experiments are given. The subjective and objective data of Chapter 4 are analysed in Chapter 5 (objective 2), while in Chapter 6 an attempt is made to derive an objective quality measure for perceptual quality from the measurement results (objective 3) by means of a regression model that describes subjective quality with objective evaluation measures. The last Chapter contains the conclusions of the report.

[3]All described algorithms are used, except the Recursive field and frame-based motion compensated noise reduction algorithms

Chapter 2

Description of noise reduction algorithms

The many noise reduction algorithms known from literature can be distinguished from one another by making a differentiation of their specific properties. The properties comprise for instance, choices of the pixels to use in the filtering action and the number of pixels that are taken into account. The next list indicates the most important properties on which the noise reduction algorithms can be distinguished [1]:

- Selection of the filter support.
- Selection of the support size.
- Noise reduction algorithm structure.
- Selection of the feature to be processed.
- Selection of the neighbourhood and adaptivity.
- Selection of statistics.
- Complexity and other hardware constraints.

In the next paragraphs, these characteristics will be explained in more detail.

2.1 Selection of filter support

The filter support is the set of pixels surrounding the pixel being filtered[1], that are candidates for the neighbourhood selection (Section 2.5). Different choices can be made on the number of dimensions along which the image is being filtered (one, two or three), the type of relation within the filter support (spatial or temporal) and the possible incorporation of motion trajectory information. Figure 2.1 gives some typical examples of filter support choices. In this figure, 16 different pixel positions are given of two consecutive fields and some possible

[1] In temporal filtering usually one other pixel is being selected for the filter support.

filter support choices. The black square is the current pixel position and the grey squares are the additional filter support pixels.

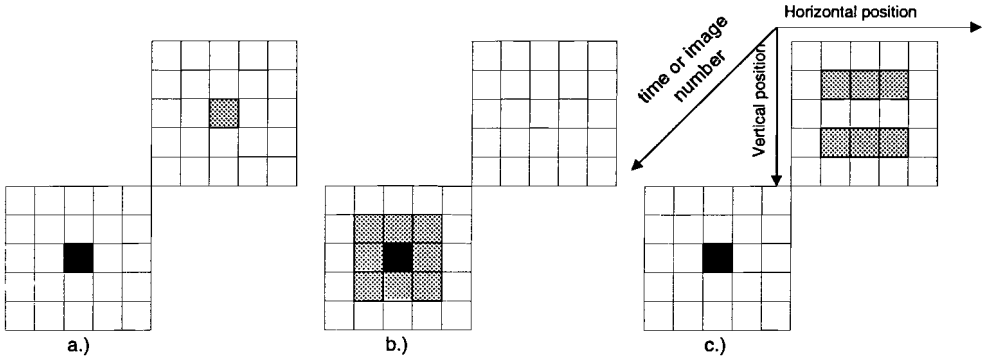

Figure 2.1: *Three examples of filter support choices. Example a.) is 1-D temporal filter support, b.) is a 2-D spatial filter support and c.) is a 3-D spatio-temporal filter support.*

Motion trajectory information is velocity information of the pixel being filtered. It indicates the x- and y-direction in which the information at a certain pixel-position is moving from one image to another. Motion trajectory information incorporation is indicated as 'MC', meaning Motion Compensation. The resulting possibilities are:

- 1-D,
  - spatial, i.e. horizontal, or vertical filtering,
  - temporal, or field- or frame-based temporal filtering and
  - MC-temporal, which means field- or frame-based temporal filtering along the motion trajectory.
- 2-D,
  - spatial, i.e. combined horizontal and vertical filtering,
  - spatio-temporal, i.e. combined horizontal or vertical and field- or frame-based temporal filtering and
  - MC-spatio-temporal, or combined horizontal, or vertical, and field- or frame-based temporal filtering along the motion trajectory.
- 3-D,
  - spatio-temporal, i.e. combined horizontal and vertical and field- or frame-based temporal filtering and
  - MC-spatio-temporal, i.e. combined horizontal and vertical and field- or frame-based temporal filtering along the motion trajectory

In spatial noise reduction algorithms, mostly, the filter support is chosen symmetrically around the pixel that is being filtered. This choice can be explained, by comparing the correlation of pixels that are near to each other and to that of more spread pixels. The correlation of the former is, in general, larger than that of the latter. This property is also important in the selection of the support size, as explained in Section 2.2.
In the case of recursive temporal filtering, only the current pixel and a time-delayed version of the pixel are being used (See *a)* in Figure 2.1). In temporal filtering, motion is the main cause that the correlation among filter support pixels decreases with larger time differences between pixels.

2.2 Selection of support size

The selection of the support size has a direct effect upon the effectiveness of the noise reduction filter. In general, a larger support size increases the noise reduction rate. However, the distortion due to the noise filtering action is often inversely correlated with the noise reduction power of the filter, i.e. when the support size increases the potential distortion increases. This effect can be explained by looking at the correlation of pixels with respect to the pixel that is being processed. The larger the filter support, the farther away are the pixels of the filter support (on average) from the pixel being processed and the less the correlation becomes among the pixels used in the filter support. Preferably, the statistical characteristics of the noise and the image pixels should differ as much as possible in the filter support to have a maximum noise reduction effect. The number of operations to filter an image using a spatial filter, where all filter support pixels are averaged together is very large.

2.3 Noise reduction algorithm structure

Noise reduction algorithms can be built using two different kinds of structures:

- Finite impulse response (FIR), i.e. the filter inputs are only unfiltered feature values.
- Infinite impulse response (IIR), i.e. among the filter inputs are also filtered feature values.

FIR-type filters are mostly used for spatial filters, because FIR-filters have always a finite impulse response and, therefore, do not visibly degrade the image content. When using the same feature input values, the recursive filters or IIR-filters have a better noise suppression than FIR-filters. This is the reason that the recursive filter structure is often used for temporal filtering, where expensive (frame or field) memory has to be used. Another reason is that horizontal recursions needed for a spatial implementation are difficult to implement in hardware, because of pipelining constraints. The negative effects of the long-tailed and asymmetrical impulse response of a recursive filter can be avoided by special measures, such as motion detection (Section 2.8.3) and motion compensation (Section 2.5).

2.4 Selection of the feature to be processed

There are two features that are usually chosen for filtering, namely *luminance* (or *grey-value*) information and *chrominance* (or *colour*) information. As explained earlier (Chapter 1), the most important noise type considered in this report is white Gaussian noise. When the video signal arrives at the consumer application, after being 'degraded' with noise (and possible other distortions), noise is present over the full frequency spectrum of the video signal. This means that noise is present in both luminance and chrominance information of the signal. It has been found that the noise in the luminance signal is much more annoying than the noise in the chrominance signal in PAL-signals (and in NTSC-decoded video as well) [2]. This can be explained because the luminance signal has a larger bandwidth than the chrominance signal. The total noise power that is accumulated onto the luminance signal is larger than on the chrominance signal. The inference is that white noise deteriorates the luminance signal more. Furthermore, the human eye is less sensitive for chrominance information, which was one of the reasons to use less bandwidth for chrominance signal in the first place. Therefore, it is easier to remove the noise from the chrominance signal, without the resulting filtering distortions being annoying to the human eye.

2.5 Selection of neighbourhood and adaptivity

In many cases, it is advantageous to select a subset of the filter support to improve the correlation among the pixels used for filtering. This subset is called the neighbourhood. The number of pixels in the neighbourhood is, therefore, smaller than or equal to the number of pixels in the filter support. The most important neighbourhood selection methods are:

- Standard neighbourhood (Figure 2.2).

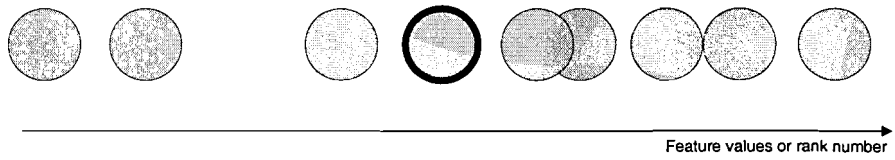

Figure 2.2: *The standard neighbourhood*

The standard neighbourhood consists of all pixels of the filter support. The grey pixels in Figure 2.2 are the pixels chosen for the neighbourhood. The pixel with the dark edge is the pixel being filtered.

- K-nearest neighbourhood (Figure 2.3).

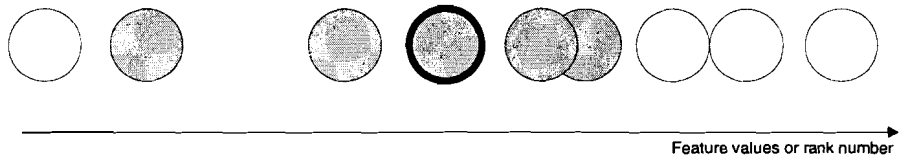

Figure 2.3: *The K-nearest neighbourhood*

In the K-nearest neighbourhood selection method, the pixel feature value (i.e. luminance or chrominance) is ranked by their difference with the value of the pixel to be processed. The K pixels with the smallest differences are chosen for the neighbourhood. An example of the K-nearest selection method with K = 4 is shown in Figure 2.3, where the grey pixels are the neighbourhood selected pixels and the white pixels are the remaining filter support pixels that are not selected.

- Sigma-nearest neighbourhood (Figure 2.4).

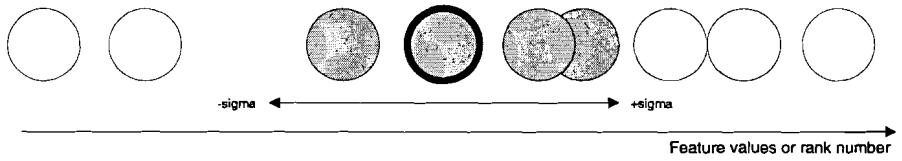

Figure 2.4: *The sigma-nearest neighbourhood*

This neighbourhood consists of all the pixels that have a pixel feature difference with respect to the value of the pixel being processed smaller than a certain threshold, σ. Sigma is correlated to some statistic property of the image, mostly, the amount of noise in the image, which will be explained in Section 2.6.

- Symmetric nearest neighbourhood (Figure 2.5)

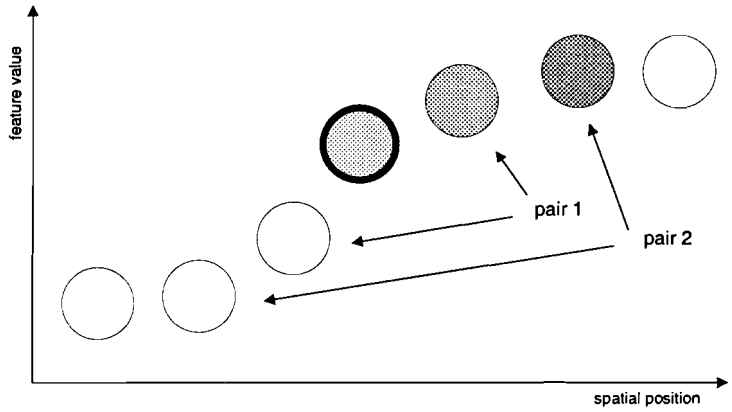

Figure 2.5: *The symmetric nearest neighbourhood*

This neighbourhood selection method chooses from each symmetrically oriented pair of neighbouring pixels in the filter support the pixel with a value that is closest to the pixel being processed. This selection is done for a number of pairs of pixels that is limited by the filter support size.

The difference between the four kinds of neighbourhood selection mthods is that the standard, the K-nearest neighbourhood and the symmetric nearest neighbourhood selection methods always select a fixed number of pixels, whereas in the sigma-nearest neighbourhood method the selected number of pixels can differ depending on the characteristics of the feature values of the pixels in the filter support. Apart from this adaptation of the filter to the *local* statistics, the filter can also be made adaptive to *global* statistics, such as the noise level measured every one or two images.

2.6 Selection of statistics

While in the previous section neighbourhood selection methods were described, here combination methods of the selected pixels of the neighbourhood are given. The method most used to combine the selected pixels of the neighbourhood is, taking the mean or weighted mean of the pixels. The weights in the weighted mean can, for instance be used to include edge or distance information in the filtering. The most important methods in which weights are assigned to pixels are:

- (Weighted) mean. (Figure 2.6)

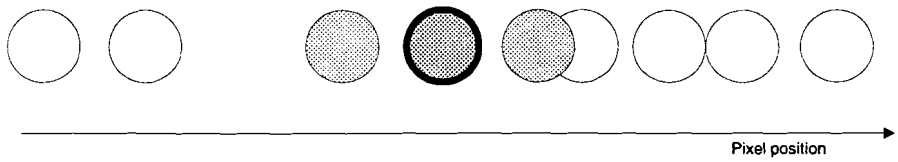

Figure 2.6: *(Weighted) Mean statistics assignment*

In this method the pixels are combined by multiplying the feature values of the neighbourhood by weights. If the weights are the same, this method takes the mean value of the feature values of the pixels in the neighbourhood. Also a weighting can be applied which, for instance gives a larger weight to feature values of pixels that are near to the pixel that is being filtered.

- Ordering of feature value. (Figure 2.7)

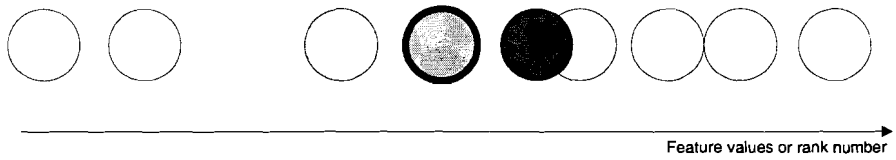

Figure 2.7: *Order statistics weights assignment*

The pixels from the neighbourhood are ordered on feature value and consecutively multiplied by certain weights that give a certain preference to a certain rank of a pixel. The resulting filter is called an order statistical (OS) filter. An important example is a median filter (Figure 2.8),

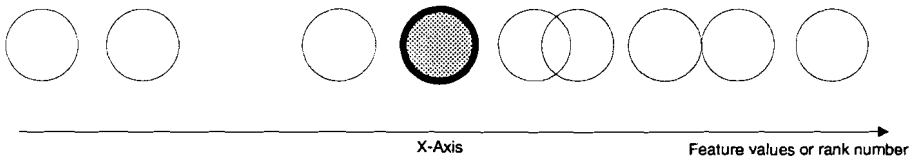

Figure 2.8: *Example of OS-filter: median filter*

which assigns the unity weight to the pixel that has a feature value that equals the median of the available feature values in the filter support.

- Ordering of differences. The difference with the OS-filter lies in the fact that the pixels in the neighbourhood are ordered by the absolute difference of the feature value with that of the pixel being filtered, instead of

the feature value itself. The resulting filter is called a differential order statistical (DOS) filter. (Figure 2.9)

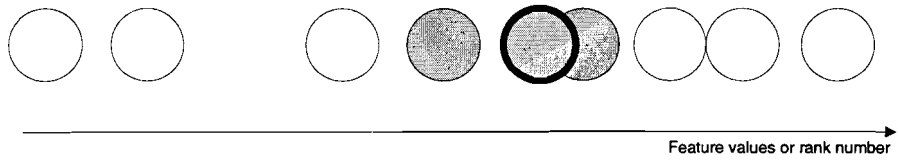

Figure 2.9: *The ordering of differences statistics*

2.7 Complexity and other hardware constraints

Noise filters in consumer applications usually have to meet severe demands in terms of cost-effectiveness. There are certain generally applicable rules that can be used to increase the cost-effectiveness of noise reduction algorithms.

- The use of the recursive filter structure can reduce the use of memory. This is because the recursive filter has a better noise suppression than the FIR filter structure (with the same number of delay/memory elements). Therefore, the support size can be chosen smaller, while maintaining the same noise reduction. This is especially advantageous in the filtering in the temporal dimension. The field or frame memory used here is quite expensive. In the filtering of the spatial dimensions, the memory demands are less stringent.

- The use of filter coefficients that are integer powers of two. Using these coefficients will speed up the processing speed, because slow multiplications can be substituted by fast shift operations. In the case of coefficients that are one, no multiplications are necessary. This property can be used in a spatial averaging filter.

2.8 Noise reduction used in consumer applications

In the following paragraphs, a selection of the most relevant consumer video noise reduction algorithms is described that filter noise from luminance information. The presented filters are suitable for implementation in consumer TV-applications. They can comply to the earlier posed demands on complexity and memory usage, because all presented filters are actually implemented in TV-systems or are suitable for TV-implementation. It has to be noted here, that the perception adaptive temporal TV-noise reduction algorithm and the recursive temporal frame-based motion compensated noise reduction algorithm need two field memories, or one frame memory. This, in contrast to the other temporal filters that need only one field memory. In terms of memory usage,

these demands can be too high if no significant additional subjective improvement can be obtained and if no other (more stringent) demands are posed on memory by other parts of the TV-system. The presented algorithms can be divided into three categories:

1. Spatial noise reduction algorithms. The filters that are described, are the edge preserving spatial noise reduction algorithm [3–6] and the Spatial Weighted Aperture Noise reduction algorithm (SWAN) [7].

2. Temporal noise reduction algorithms. Discussed are the recursive temporal field-based motion compensated noise reduction algorithm, the recursive temporal frame-based motion compensated noise reduction algorithm, the perception adaptive temporal TV-noise reduction algorithm [3–6], the Dynamic Noise Reduction (DNR) algorithm [1] and the SGS Thomson fuzzy temporal noise reduction algorithm [8, 9].

3. Combined (spatial and temporal) noise reduction algorithms. The SGS Thomson temporal median filter [8, 9] is the only filter that falls into this category.

All filters are characterised by giving the details with respect to the filter characterisation choices as described in the previous paragraphs.

2.8.1 Edge preserving spatial noise reduction algorithm

This noise reduction algorithm [3–6] uses a filter support of nine pixels centred around the pixel being processed.

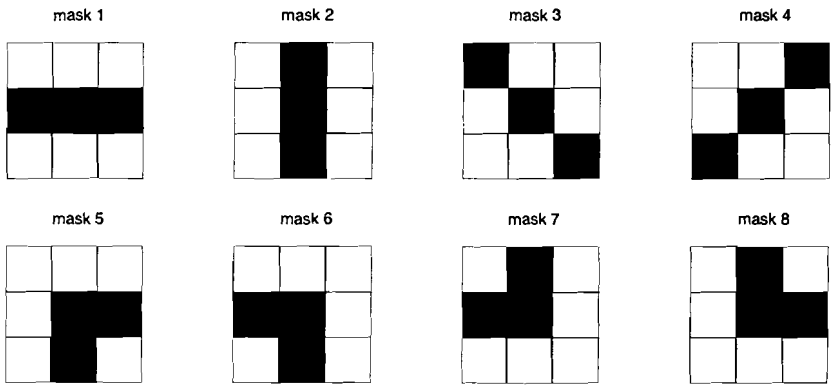

Figure 2.10: *The eight potential neighbourhoods used in the edge preserving spatial noise filter*

The filter chooses from eight different potential neighbourhoods within this support. Each potential neighbourhood consists of three pixels. In Figure 2.10 the used potential neighbourhoods are shown. The black pixels are the ones chosen for the potential neighbourhood, while the white pixels are neglected. The selection of the neighbourhood from the eight possible ones is done by using the following procedure:

1. Calculate for all eight potential neighbourhoods the high-pass output. The high-pass output is obtained by using a high-pass filter with coefficients (-1, 0, -1), where the weight 0 is assigned to the center pixel and the weight -1 to the other pixels in eight different ways (Figure 2.10).

2. Select the neighbourhood with the minimum high-pass output. This selection is very likely to be the neighbourhood that distorts the details of the image the least and attenuates the noise the most.

3. Low-pass filter the pixel with its selected neighbourhood. The filter coefficients are $(\frac{1}{4}, \frac{1}{2}, \frac{1}{4})^2$, with a filter weight $\frac{1}{2}$ for the center pixel and $\frac{1}{4}$ for the other two pixels of the selected negihbourhood.

The high-pass filtering and subsequent selection of the neighbourhood are a selection of statistics method, that is used to select pixels for filtering in such a way that the maximum amount of detail is preserved, at a certain noise suppression rate. The number of potential neighbourhoods that is used adjusts the trade-off of the filter between two properties. First, large noise reduction (achieved with few filter masks) and, second, large detail-preservation ability (achieved with many filter masks) in structured areas of an image. The most effective noise reduction in homogeneous areas can be obtained by using a standard neighbourhood (of three pixels), while the best detail-preservation in structured areas can be obtained with a K-nearest neighbourhood selection method (with K=2), which has 28, i.e. $\binom{8}{2}$, possible filter-configurations for a 3x3 filter support. The number of masks chosen leads to a theoretical noise reduction of 2 dB in homogeneous areas. It has been found in simulations that the overall noise reduction was about 1 to 2 dB, as explained in [3]. The performance of this filter is adjusted to be used in cascade with the subband based temporal noise reduction filter of Section 2.8.6. This temporal filter has motion detection that causes the filter to switch off on edges in structured areas. In these areas the filter performance is, therefore, worse than in homogeneous areas. The spatial filter is adjusted to compensate for this lack of performance in structured areas. Where the temporal filter lacks in performance, the spatial filter gives some extra noise reduction. The spatial filter has an extra beneficial effect, namely that the motion detector of the temporal filter is more accurate because the input is less noisy, resulting in an extra gain in noise suppression. In [3] the noise reduction has been tested in simulations and the results are given in Table 2.1. In this table, the noise evaluation measures *SNR* and *SNRI* used that will be explained in Chapter 3.

2.8.2 Spatial Weighted Aperture Noise reduction algorithm

The SWAN algorithm [7] is a spatial filter which integrates both noise reduction and peaking. Cascading of noise reduction and peaking has not an optimal effect [7], due to the contradictory nature of these two actions. Noise filtering is, namely, a low-pass filtering operation, while peaking is in general a high-pass

[2] In the high-pass filter only the relative output is of importance, while in the low-pass filter normalisation needs to be done.

Table 2.1: *Performance results of the edge preserving spatial noise filter [3]*

Kind of sequence	Noise improvement [dB *SNRI*]
Zoom 25 dB *SNR*	1.69(2.55)
Zoom 30 dB *SNR*	1.43(0.99)
Static sequence 25 dB *SNR*	1.71(2.3)
Natural sequence 25 dB *SNR*	1.63(1.23)

Between brackets in the right colomn is the noise reduction added to the performance of the spatial filter when comparing temporal filtering only, and the cascaded filtering of the spatial filter and the subband based temporal noise reduction filter of Section 2.8.6.

filtering operation. SWAN separates the pixels of the neighbourhood into three categories, i.e. pixels that are used for the noise filtering action, pixels that are used for the peaking operation and pixels that are not used to determine a filtering action. The filtering operations are assigned in a mutually exclusive way and, thus, inefficiency due to cascading is avoided.
The SWAN algorithm uses a 2-D window of five lines by 17 pixels (Figure 2.11). From every line of this window, five pixels are selected for the filter support. In total 25 pixels are used in the filter support (the selection method will be described further in this section). The distance, both horizontal and vertical, between the pixels influences the frequency response of the filter: a larger distance between the pixels increases the low-frequency noise suppression, while it lets the high-frequencies pass better (vice versa for a smaller pixel distance). A horizontal distance of 4 pixels was found to give acceptable results [7]. Furthermore, an increasing positive horizontal offset from the (horizontal) position of the center pixel is chosen for the two upper and bottom lines of pixels to prevent ringing effects in the filtered video, as is indicated in Figure 2.11.

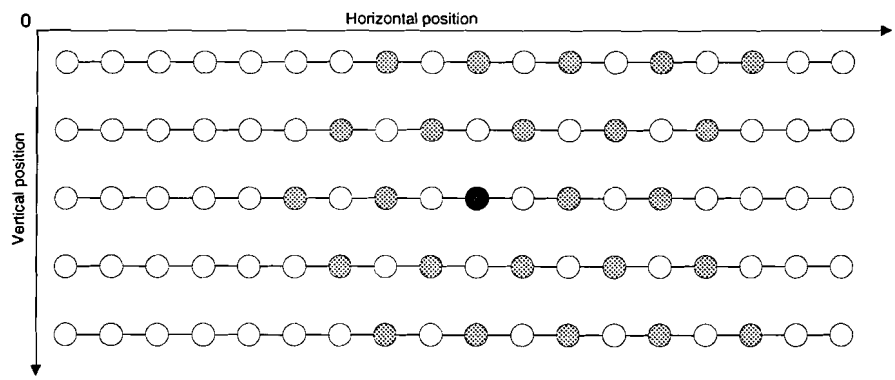

Figure 2.11: *The filter support of the SWAN algorithm*

For all 25 pixels the luminance difference of the pixel to the central pixel is calculated. The calculation of the pixel weights is done using five thresholds. The assignment of the weights is visualised in Figure 2.12. As can be seen in this figure, positive (level 1 to 3), negative (level -1) and zero weights can be assigned to the pixels. Small luminance differences probably result from noise and need to be noise filtered, while large pixel differences more likely result from spatial detail and are used as candidates for peaking. The pixel differences that are used for peaking can be adjusted in a more elaborate way. This has been found useful to prevent the amplification of impulse noise. The thresholds $thr4$ and $thr5$ can be used to select the right peaking range for the pixel differences. Threshold $thr4$ is used to have an offset between the noise reduction range and the peaking range. This can be helpful to prevent the amplification of peaks of noise. Threshold $thr5$ is used to prevent the amplification of peaks resulting from impulse noise.

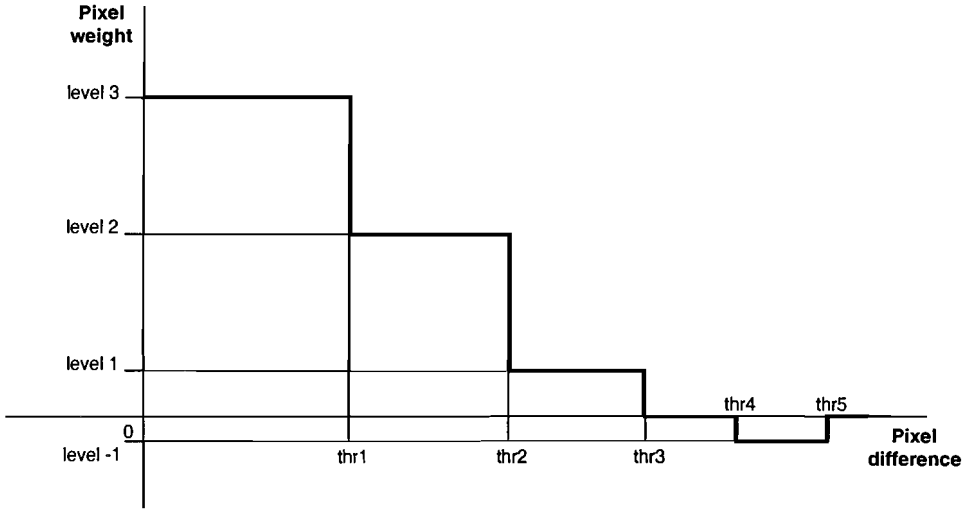

Figure 2.12: *The calculation of weights assigned to the pixel differences*

The thresholds are changed in accordance with the noise variance of the video. The noise level is automatically measured by using a noise estimator, as described for instance in [1, 10, 11]. SWAN can be seen as a combination of a DOS-filter (different rankings are assigned different values) and a sigma-filter (noise variance is the measure that is used to select the pixels that are used as the neighbourhood). Statistical selection in the SWAN filter is used in the way that weights are assigned to the pixel value differences (and corresponding pixels): small to middle large (positive) weights are assigned to large to middle large pixel differences, while fixed (negative) values are assigned to large pixel differences. The sum of the weighted pixels represents the combined effect of noise reduction and peaking in a mutually exclusive way for all pixels of the filter support. The discrimination between the three different kinds of pixels is done in the Discriminating Averaging Filter (DAF), that is shown in the SWAN diagram in Figure 2.13.

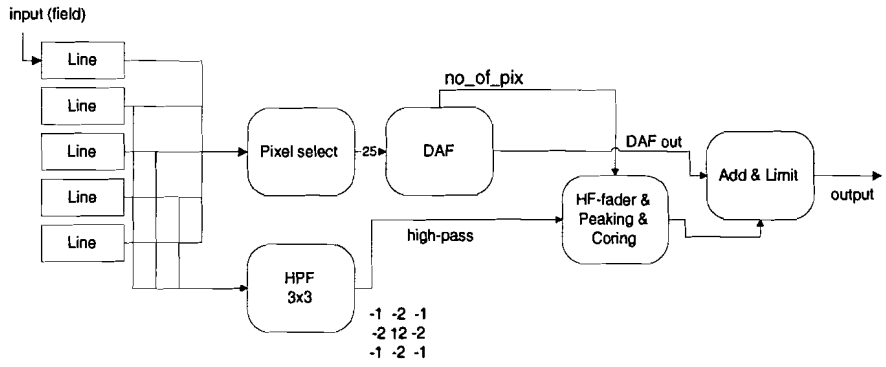

Figure 2.13: *Block diagram of SWAN*

The output of DAF is the weighted sum of pixels and the number of pixels (smaller or equal to 25) that have been assigned positive weights (i.e. that are used for noise reduction). This last number is used to adjust the amount of high-frequency bypass through the use of the HF-fader. The more pixels are used for noise reduction, the larger the gain of the high frequencies in the HF fader. The high frequencies are obtained by using a high-pass filter on the video signal before it enters the DAF. The high-frequency signal is processed subsequently by using peaking and coring. The DAF output and the high-frequency signal are added to get the output of the SWAN filter. The limiter assures that the output of SWAN is within the correct boundaries.
The performance in SNRI of the SWAN filter is given in Table 2.2 for four different sequences (*Renata*, *Circle*, *Teeny* and *Car & Gate*). Especially in very noisy video with large homogenous areas (for instance in the sequence *Teeny* that has an unweighted SNR of 26 dB), the performance is exceptionally good.

2.8.3 Recursive temporal field-based motion compensated noise reduction algorithm

The recursive temporal field-based motion compensated noise reduction algorithm uses a two pixel motion compensated filter support (Figure 2.14).

Table 2.2: *Performance results of the SWAN filter [7]*

Sequence [dB SNR]	Noise improvement [dB SNRI]
Renata	
38	0
32	0.7
26	1.8
20	3.2
Circle	
38	1.8
32	2.8
26	3.8
20	4.1
Teeny	
38	1.0
32	2.8
26	4.2
20	5.4
Car & Gate	
38	0
32	0.7
26	1.8
20	3.2

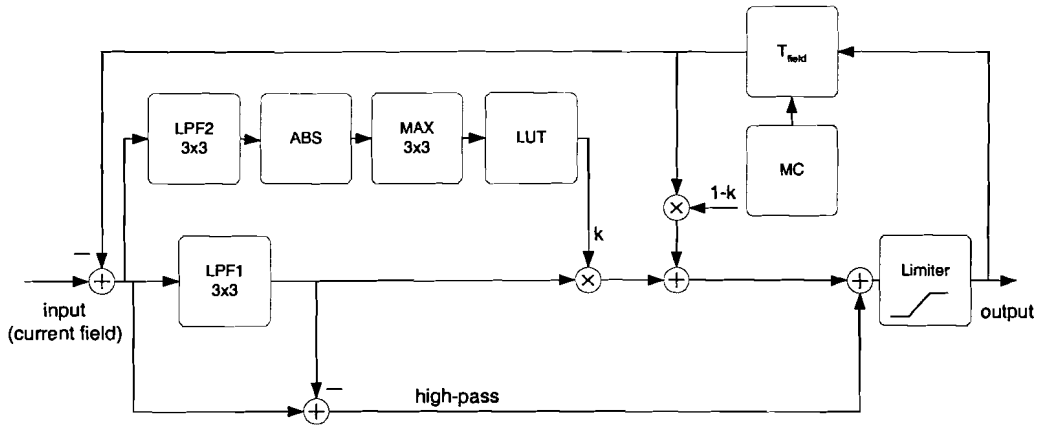

Figure 2.14: *Recursive temporal field-based motion compensated noise reduction algorithm. LPF stands for low-pass filtering, ABS stands for the absolute value operation, MAX stands for the maximum operation, LUT stand for look-up-table, MC means motion compensation and T_{field} is a field-time delay.*

It uses motion detection as an adaptive neighbourhood selection method to prevent loss of detail in moving areas of an image. The motion compensation uses sub-pixel accuracy in both the horizontal and the vertical domain. Sub-pixel accuracy means that when a motion vector points at a position in between pixel positions the luminance value is calculated from the weighted average of the surrounding four pixels, with weights inversely relative to the distance of these pixels to the true motion vector position. High video frequencies are more susceptible to distortion artefacts caused by the noise filter. As will be seen in Chapter 3 of this report, the high noise frequencies are much less annoying than low noise frequencies. The high-frequency bypass, therefore, can give a perceptive improvement. The diagram of the filter is given in Figure 2.14. The input field is low-pass filtered in LPF1 and the high-pass frequencies are obtained by subtracting the low-frequency components from the original field. The motion detection part is shown in the upper part of Figure 2.14. First, the input field is subtracted from the output signal of the temporal filter. This result is low-pass filtered by LFP2, the absolute value is taken and the maximum value is taken over a 3x3 window. Finally, the look-up-table (LUT) translates the maximised value into the k value. The k value is used as the coefficient in the recursive temporal filter, according to Equation 2.1.

$$G_F(\vec{x},n)_{LF} = (1-k)G_F(\vec{x},n-1)_{LF} + kG(\vec{x},n)_{LF} \qquad (2.1)$$

where $G(\vec{x},n)$ is the unfiltered input field at field number n, $G_F(\vec{x},n)_{LF}$ is the low-pass component of the filtered field and k is the coefficient that adjusts the noise suppression rate of the filter.
The frequency response characteristics of a field-based recursive filter as a function of k can be found in Figure 2.15. In this figure, deinterlacing effects[3] have

[3]Deinterlacing effects have a low-pass effect on the vertical frequency content of an image,

not been taken into account and no high-frequency bypass is used.

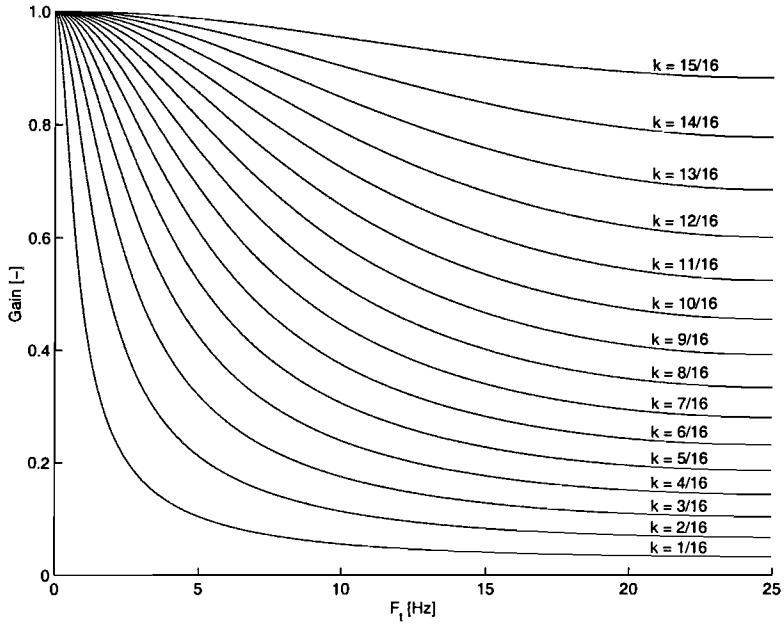

Figure 2.15: *Amplitude-frequency response of a field-based recursive filter as function of k without high-frequency bypass*

LUT can be used to adjust the filtering action to the noise strength and the motion content. Typical LUT characteristics are given in Figure 2.16. In Figure 2.16, characteristic a) is very sensitive for input changes, it switches the noise filter quickly off, while the b)-labelled characteristic is much more gradual in reducing the noise filter's effectiveness. The values for k of 0 and 1 have a special meaning: $k = 0$ means that the previous field pixel information is the output, $k = 1$, means that the current field information is passed to the output and the filter is switched off. The choice for a specific characteristic depends on the properties of the motion detector and the video that is being filtered. In general, a few specific curves are chosen that are found to be the best from perception tests or the experience of the designer. The consumer can select one curve, taking into account the noise and movement properties of the video.

which could result visually in loss of vertical detail. See for instance [1] for an example of the effects of a simple line-averaging deinterlacing on the vertical frequency response.

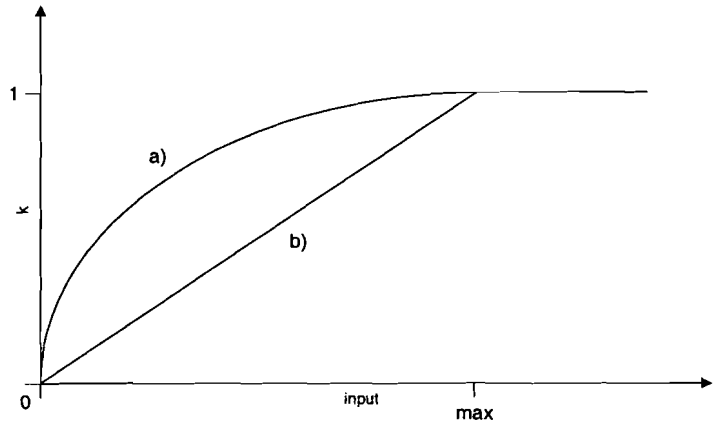

Figure 2.16: *Examples of k-characteristics*

The total output $G_F(\vec{x}, n)$ of the filter can be written as

$$G_F(\vec{x}, n) = (1 - k)G_F(\vec{x}, n - 1)_{LF} + kG(\vec{x}, n)_{LF} + G(\vec{x}, n)_{HF} \tag{2.2}$$

where $G(\vec{x}, n)$ is the field at field time position n and $G_F(\vec{x}, n)_{LF}$ and $G_F(\vec{x}, n)_{HF}$ are the low-pass and the high-pass component of the filtered field.
The value of k determines the noise suppression rate, which can be calculated from the total noise variance resulting from two independent noise sources, as in Equation 2.3.

$$\sigma_s = {\sigma_1}^2 + {\sigma_2}^2 \tag{2.3}$$

Assuming that the two noise inputs of the recursive filter act as two independent noise sources, it follows that:

$${\sigma_{out}}^2 = k^2{\sigma_{in}}^2 + (1 - k)^2{\sigma_{out}}^2 \tag{2.4}$$

Rewriting this equation gives the noise suppression rate, defined as:

$$\frac{{\sigma_{in}}^2}{{\sigma_{out}}^2} = \frac{1 - (1 - k)^2}{k^2} \tag{2.5}$$

In the case of a high-frequency bypass, only the low-frequency video input is filtered by the recursive filter, inputs to the filter are $G_F(\vec{x}, n)_{LF}$ and $G(\vec{x}, n)_{LF}$. The output noise variance can be divided into low and high frequencies, where the low-frequency part of the video is filtered according to Equation 2.4. The total noise variance, ${\sigma_{out}}^2$, can be written as:

$${\sigma_{out}}^2 = k^2{\sigma_{LF_{in}}}^2 + (1 - k)^2{\sigma_{LF_{out}}}^2 + {\sigma_{HF}}^2 \tag{2.6}$$

The high frequencies are a part of the total noise variance, ${\sigma_{in}}^2$, and can, therefore, be written as:

$${\sigma_{HF}}^2 = c\ {\sigma_{in}}^2 \tag{2.7}$$

where c is defined as

$$c \equiv \frac{{\sigma_{HF}}^2}{{\sigma_{in}}^2}, \quad c \in \{0, 1\} \tag{2.8}$$

Table 2.3: *Noise reduction gain in a first order recursive temporal filter as function of coefficient k with no high-bypass*

k	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1
SNRI [dB]	0.87	1.76	2.69	3.68	4.77	6.02	7.53	10.0	12.79
SNRI(HF-bypass) [dB]	0.46	0.97	1.55	2.22	3.01	3.98	5.23	6.99	10.0

The noise suppression rate including the high-frequency bypass can be written as:

$$\frac{{\sigma_{in}}^2}{{\sigma_{out}}^2} = \frac{(1-c)(1-(1-k)^2)+c\ k^2}{k^2} \tag{2.9}$$

Taking a simple, but often used, high-pass filter with coefficients $(-\frac{1}{4},\frac{1}{2},-\frac{1}{4})$, then c is approximately 0.5. In this case, the noise reduction rate becomes

$$\frac{{\sigma_{in}}^2}{{\sigma_{out}}^2} = \frac{1}{2}\ \frac{(1-(1-k)^2)+k^2}{k^2} \tag{2.10}$$

In Table 2.3, the *SNRI* in dB is given as function of k for the two important cases: high-frequency (HF) bypass and no HF-bypass.

2.8.4 Recursive temporal frame-based motion compensated noise reduction algorithm

The recursive temporal frame-based motion compensated noise reduction algorithm (Figure 2.17) uses the same principles as the field-based version. The only difference of the filter is that the two pixels of the filter support are a frame time apart, rather than a field time. The total output $G_F(\vec{x},n)$ of the filter can be written as

$$G_F(\vec{x},n) = (1-k)G_F(\vec{x},n-2)_{LF} + kG(\vec{x},n)_{LF} + G(\vec{x},n)_{HF} \tag{2.11}$$

where $G(\vec{x},n)$ is the unfiltered input field at field number n and $G_F(\vec{x},n)_{LF}$ and $G_F(\vec{x},n)_{HF}$ are the low-pass and high-pass components, respecively, of the filtered field.
The amplitude-frequency response of a frame-based recursive filter as function of k can be seen in Figure 2.18. This characteristic can be deducted from the amplitude-frequency characteristic of the field-based filter by considering the frame-based filter as a down-sampled version of the field-based filter (Figure 2.15) with a factor two. The main difference between the two filter characteristics is that the frame-based recursive filter has a steeper fall-off frequency for the low-frequency range, while it passes high frequencies that are blocked by the field-based filter.

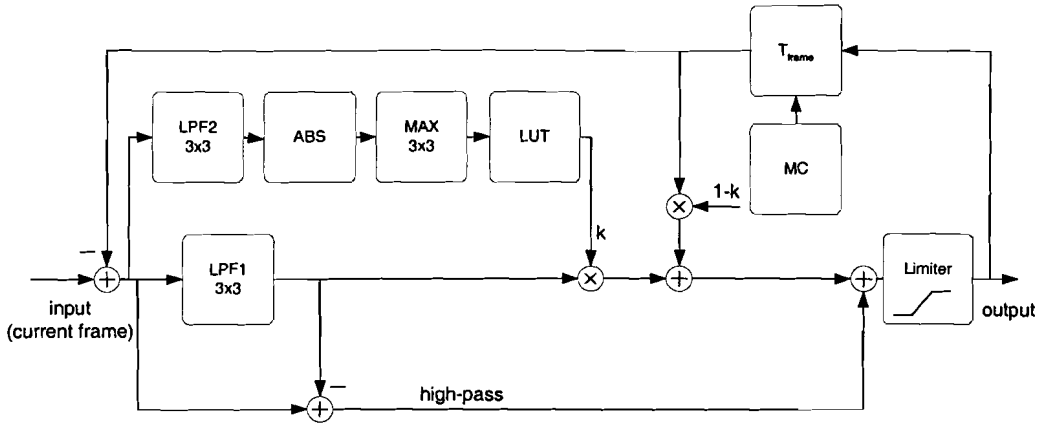

Figure 2.17: *Recursive temporal frame-based motion compensated noise reduction algorithm. LPF stands for low-pass filtering, ABS stands for the absolute value operation, MAX stands for the maximum operation, LUT stand for look-up-table, MC means motion compensation and T_{frame} is a frame-time delay.*

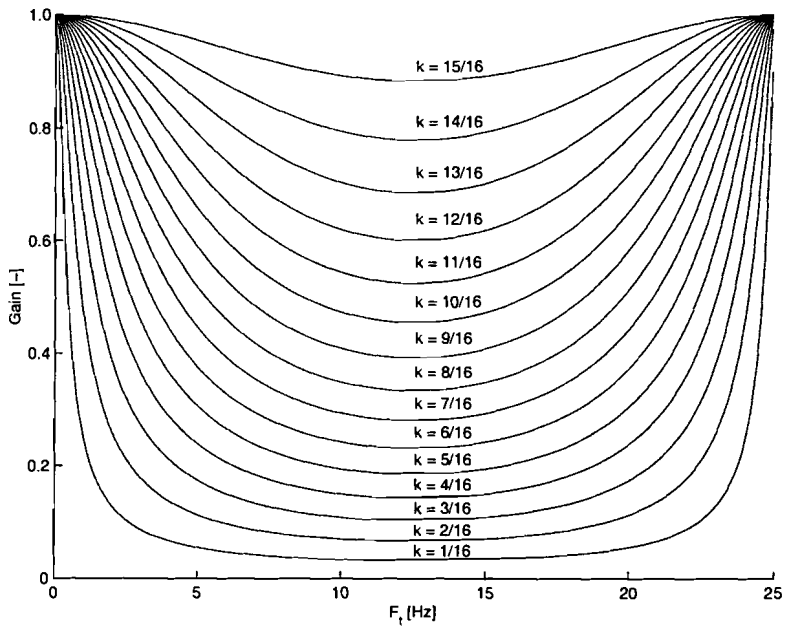

Figure 2.18: *Amplitude-frequency response of frame-based recursive filter as function of k*

2.8.5 Dynamic Noise Reduction

The Dynamic Noise Reduction (DNR) algorithm [1] uses a two pixel horizontal and vertical motion compensated filter support with a temporal field-based recursive filter. It uses motion detection as a form of adaptive neighbourhood selection. The DNR algorithm had been implemented earlier also in the Falcon IC that is used in the current generation televisions, and will be referred to in this report as Falcon. Before Falcon, the algorithm was also used simply under the name DNR. This first DNR did not have a high-frequency bypass nor motion compensation. In the more recent Falcon implementation horizontal motion compensation was added to DNR as well as a high-frequency bypass. Condor is a software simulation model that simulates part of the new IC for video processing that will, among others, be used in the next generation Philips televisions. The SWAN and DNR algorithms are implemented in this software platform. The DNR version that is used in Condor is shown in Figure 2.19.

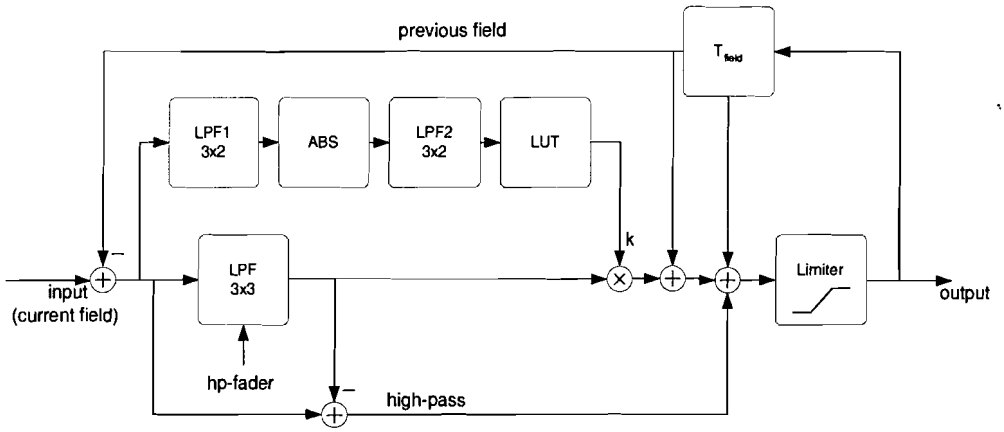

Figure 2.19: *Dynamic Noise Reduction (DNR) algorithm as used in Condor*

From this figure the working of the DNR algorithm can be explained. The output signal of the Condor DNR with a delay of one field time is subtracted from the input signal and low-pass filtered in the 3x3 low-pass filter (LPF). The low-pass signal is subtracted from the input signal to LPF1 to obtain the high-frequency components. The amount of high-pass frequencies can be adjusted with the HP fader. The motion detection is done by low-pass filtering the input signal to LPF by using LPF1. From the output of LPF1 the absolute value is taken. This signal is filtered again in LPF2 and by using a look-up table (LUT) the value for k is obtained. After adding all the intermediate results that are indicated in Figure 2.19, the filter output $G_F(n)$ can be written as

$$G_F(\vec{x},n) = (1-k)G_F(\vec{x},n-1)_{LF} + kG(\vec{x},n)_{LF} + G(\vec{x},n)_{HF} \qquad (2.12)$$

where $G(\vec{x},n)$ is the field at field time position n and $G_F(\vec{x},n)_{LF}$ is the low-pass component of the filtered field, $G_F(\vec{x},n)_{HF}$ the high-pass part of the filter

input field and k is used to set the noise suppression rate.
The size of the low-pass filters in the motion detection path and the size of the low-pass filter in the signal path might seem odd at first sight, however, they have been carefully matched for time delays and operation in accordance with motion estimation blocks that are calculated in other parts of the Condor software platform.

2.8.6 Perception adaptive temporal TV-noise reduction

This filter [3–6] uses a three pixel filter support. It has a recursive field-based filtering structure, with a motion detector, as a means of adaptive neighbourhood selection to prevent loss of detail in moving image areas. In the filter both temporal filtering on field and frame basis is used. Recursive temporal filtering on field-basis has the disadvantage that in consecutive fields the same pixel-line positions are not available, due to interlacing. In Figure 2.20, this property of interlaced video is shown. It shows four consecutive fields of video, where the current field is represented by the dark blocked lines, whereas the white lines contain the pixels that are unavailable. When looking at a certain pixel position, for instance the black pixel, no video information is available in the adjacent fields. The adjacent field pixel can only be approximated, e.g. by using interpolation techniques on the previous field pixels. Due to this effect, the use of a field based recursive temporal filter results in a loss of vertical resolution.
The frame based recursive temporal filter has the advantage that pixels are available at exact the right position in the video, and consequently, no loss of resolution will occur here (See again Figure 2.20). However, the time difference between frames is twice as large as the time difference between fields, i.e. 40 ms instead of 20 ms. This means that moving objects can be displaced considerably in consecutive frames. The result is that the recursive filtering between two adjacent frames will be partly switched off by the motion detector in non-matching areas. This problem can be overcome by using motion compensation. One problem, however, can not be solved by motion compensation. When an object is occluded by another moving object, new image parts will appear that can not be filtered because the image information is not available in previous images (yet).

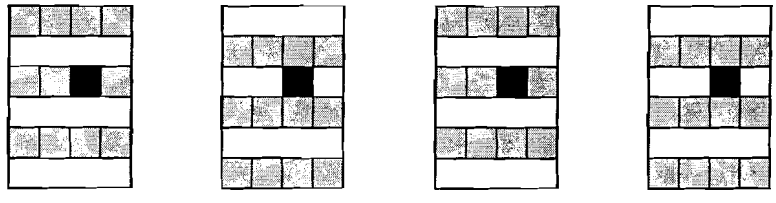

Figure 2.20: *Consecutive fields with field numbers n [-] and time t [ms]*

The disadvantages of both time basis methods can be reduced by using a subband based processing, which is adapted to perception properties of the human eye. In accordance with human perception one can divide video into two subbands: a low frequency band and a high frequency band. Processing for the low- and for the high-frequency band has distinct properties. The human eye has better spatial accuracy for high-frequency content, while the human eye is in the low-frequency part more accurate for temporal differences [4]. In case of non-moving areas for low frequencies the spatial accuracy is the most important, thus, in that case a frame-based filter is preferred. This induces the choice for a recursive temporal filter on field basis in the low-frequency spectrum with the special ability to preserve the spatial content in non-moving areas. This last property can be assured by adding a median-based deinterlacing to the low-frequency processing.
In the high frequency part of the video, recursive temporal filtering on a frame basis is used in order to preserve the full spatial resolution. Because, the temporal field-based filter only has loss of accuracy in vertical frequencies, only a division of the video spectrum into vertical low and high frequencies has to be considered. The division of the video spectrum into two subbands has the added advantage that the two filters can be individually adjusted (to comply for instance, to the weighted noise spectrum of [12–14]), so that maximum subjective gain can be reached.
The recursive filtering on field basis is described by Equation 2.13.

$$G_F(\vec{x},n)_{LF} = k_{LF}G(\vec{x},n)_{LF} + (1-k_{LF})\ median\begin{pmatrix} G_F(x,y,n-2)_{LF} \\ G_F(x,y-1,n-1)_{LF} \\ G_F(x,y+1,n-1)_{LF} \end{pmatrix} \tag{2.13}$$

where $G(\vec{x},n)_{LF}$ is the low vertical-frequency content of the nth unfiltered input field and $G_F(\vec{x},n)_{LF}$ are the low vertical frequencies of the nth output field and k_{LF} sets the noise suppression rate.

The *median*() operation is used to enhance the noise reduction in non-structured areas; low noise-frequencies that are important for human perception quality. In addition, the spatial accuracy is improved for static image content.
The frame based recursive filtering for the high-frequency part is described in Equation 2.14.

$$G_F(\vec{x},n)_{HF} = k_{HF}G(\vec{x},n)_{HF} + (1-k_{HF})G_F(x,y+1,n-2)_{HF} \tag{2.14}$$

where $G(\vec{x},n)_{HF}$ is the high-frequency part of the nth unfiltered input field, $G_F(\vec{x},n)_{HF}$ is the high-frequency part of the nth output field and k_{LF} sets the noise suppression rate.

The diagram of the total filter can be seen in Figure 2.21.

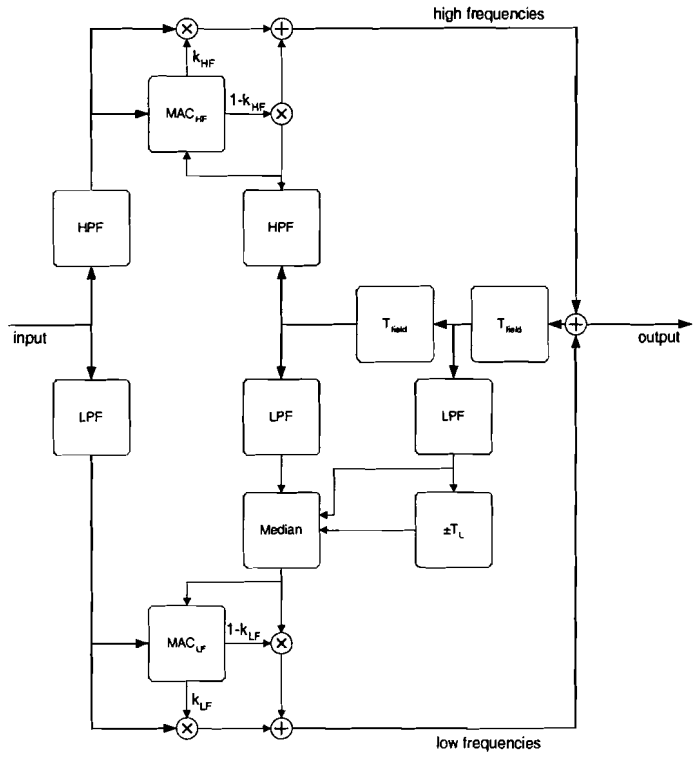

Figure 2.21: *Perception adaptive temporal TV-noise reduction algorithm*

In the upper part of the diagram, the input video signal is first split into the high vertical-frequency part and the low vertical-frequency part (in the lower part) by using a simple 3-taps LPF-filter with coefficients (1,2,1) and an orthogonal HPF-filter. In the (vertical) high-frequency processing part of the diagram of Figure 2.21, the Motion Adaptive Control unit for high vertical frequencies, MAC_{HF}, is shown. This motion detector uses an absolute value operation, a 3x3 averaging filter, a 3x3 maximum operator and a LUT for the k_{HF} calculation, in a similar way as for the version of DNR that is implemented in the Condor software platform and was described in Figure 2.19. Control parameter k_{HF} is used in the recursive filter to calculate $G_F(\vec{x},n)_{HF}$, as found in Equation 2.14. A similar process is used for the low vertical-frequency part. Here, the median-based deinterlacing is added with help of the line alternator T_L. The output of the low vertical-frequency part, $G_F(\vec{x},n)_{LF}$, is calculated according to Equation 2.13. The two filter outputs $G_F(\vec{x},n)_{LF}$ and $G_F(\vec{x},n)_{HF}$ are summed to obtain the total output of the filter.

2.8.7 Fuzzy temporal noise reduction algorithm

The filter [8,9] uses a seven pixel filter support in a field based temporal recursive filter structure (See Example c) in Figure 2.1 and Figure 2.22.).

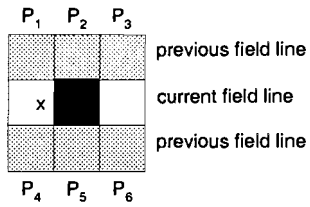

Figure 2.22: *The filter support of the spatio-temporal median filter of SGS Thomson*

In this figure the previous field lines and the current field lines are shown at the same time. Pixel x is filtered with the six different pixels numbered P_1, P_2, P_3, P_4, P_5 and P_6.
It uses two adaptive statistical neighbourhood methods:

- motion correlation (CorrMot), to preserve detail in areas with motion.
- (inversed) spatial correlation (Corr), to preserve areas with detail.

The motion correlation (CorrMot) is calculated by looking at the difference between the average spatial feature value of the previous field and the feature values of the current field x (Equation 2.15). The averaging operation $|(P_i + P_j)/2$, interpolates the missing feature value on the position of x.
The spatial correlation (Corr) is defined as the spatial difference of luminance values, as indicated in Equation 2.15.

$$\begin{aligned} CorrMot &= |(P_i + P_j)/2 - x| \\ Corr &= |P_i - P_j| \\ \\ with \;\; P_iP_j &\in \{P_1P_4,\; P_2P_5,\; P_3P_6\} \end{aligned} \tag{2.15}$$

The filter uses fuzzy logic in its processing. Fuzzy logic is basically a superset of the Boolean arithmetic, in which besides the values 0 and 1, also all intermediate values are possible and can be used in calculations. This means that Corr and CorrMot can have any value in between the predefined minimum (LOW) and maximum (HIGH) values, which are in Boolean arithmetic, respectively, 0 and 1. In addition, logical operators, like the AND-operator, can be used on these operators. The fuzzy AND-operator has one special characteristic that distinguishes its operation from the operation of the logical AND-operation:

$$\begin{gathered} If\ Corr \in \{HIGH, LOW\},\ CorrMot \in \{HIGH, LOW\}, \\ then\ (Corr\ AND\ CorrMot) \in \{HIGH, LOW\} \end{gathered} \tag{2.16}$$

The fuzzy AND-operation is also defined for intermediate values, and it converts its parameters according to a certain translation table into an (fuzzy) AND-value. The conversion table is very important to get the right characteristic for the AND-operation.

The fuzzy temporal filter uses Corr and CorrMot to calculate the recursive filter coefficient, in such a way that a low value for Corr and a low value for CorrMot results in a high value for k_{nr}, the filter coefficient that determines the noise suppression rate. This property is written in the fuzzy rule (Equation 2.17)

$$IF\ Corr\ \epsilon\ (LOW, HIGH)\ AND\ CorrMot\ \epsilon\ (LOW, HIGH),$$
$$THEN\ k_{nr}\ \epsilon\ (HIGH, LOW) \tag{2.17}$$

This rule is applied in the three directions which are possible for the three combinations of P_i and P_j, namely P_1P_4, P_2P_5 and P_3P_6. From the three calculated values the maximum value for k_{nr} is selected, i.e. the maximum possible noise reduction rate. This is the direction in which $(Corr\ AND\ CorrMot)$ is minimal. The coefficient k_{nr} is used in Equation 2.18 to calculate the output pixel. This procedure is used for all pixels x of the unfiltered input field.

$$y_{nr} = k_{nr}\frac{P_i + P_j}{2} + (1 - k_{nr})x \tag{2.18}$$

where y_nr is the output of the filter, P_i and P_j are the pixels, that give the maximum k_{nr} value and x is the luminance value of the center pixel (Figure 2.22). Loss of horizontal detail can occur when the processed pixel lies on the border of two regions with different luminance values. In this case, P_i might differ much from P_j and the fuzzy process will detect a moving pixel erroneously and introduces an edge-smoothing effect. This can be avoided by introducing a deinterlacing effect for x. The feature value x will be substituted by the weighted sum of $\frac{P_i+P_j}{2}$ and $median(P_2, P_5, x)^T$ depending on the value of k_{sm}. The coefficient k_{sm} is determined by CorrMot by a fuzzy rule (Equation 2.19).

$$IF\ CorrMot\ is\ HIGH,\ THEN\ k_{sm}\ is\ HIGH \tag{2.19}$$

The substitute value for x is calculated by Equation 2.20.

$$x_{substitute} = k_{sm}\frac{P_i + P_j}{2} + (1 - k_{sm})median\begin{pmatrix} P_2 \\ P_5 \\ x \end{pmatrix} \tag{2.20}$$

with P_2 and P_5 the luminance values of the pixels near the position of x in the vertical direction and x the luminance pixel of the center pixel (Figure 2.22). The output of the filter can be calculated by using Equation 2.18 with the new value for x substituted, found from Equation 2.20.

2.8.8 Combined spatial and temporal median filtering

The median filtering algorithm [8, 9] uses a filter support of seven pixels, with the center pixel as the pixel being filtered (See Figure 2.22 and Figure 2.1, example c).). The pixels P_1, P_2, P_3, P_4, P_5 and P_6 are used in an OS-filter with weights two and pixel x is given the weight five. The current pixel is only substituted with the output value of the OS-filter, if this output differs much from the current luminance value of x.

2.9 Conclusions

Different characteristics of noise reduction algorithms have been described. A description of filter support, support size, filter structure types, noise filter features, neighbourhood selection and adaptivity, statistics sued in filtering and complexity and hardware constraints has been given. Furthermore, spatial and temporal filters have been given and one combined spatial-temporal filter has been described. Many of these filters will be used in the objective and subjective performance evaluation described in the next chapters. The filters have been used with the settings given in Appendix D.

Chapter 3

Noise filter algorithm evaluation methods

In all phases of the design of video noise filters, information is needed to determine whether certain design choices will result in an improvement of the noise reduction algorithms. In the quality assessment of noise reduction filters, a distinction can be made between

- objective,
- subjective and
- hybrid quality measures.

While the *objective* measures are made up of a figure obtained from some formula describing luminance characteristics or signal characteristics, the *subjective* quality measures are figures that describe opinions of observers about quality of noise-reduced sequences and are obtained from a viewing experiment, called a panel test. The subjective quality measures are in general more reliable than objective measures, because they measure quality directly by gathering opinions from human observers. Although the opinion of a single human observer can be less precise, through the use of statistical methods, panel tests with a number of subjects can provide precise and reliable results. Some objective measures that describe luminance characteristics of the video, try to incorporate human perception characteristics. We shall call these measures *hybrid* quality measures. There have been numerous attempts to make such hybrid qualifiers for the assessment of images (See for instance [15]). Recently, more attempts have been made to make such quality figures for video sequences, mostly for the quality assessment of new technologies, for instance of the use of MPEG streams for video purposes [16, 17].

In the real-time use of noise reduction algorithms, a difficulty arises when one wants to evaluate the performance of such an algorithm: only the noisy (unfiltered) video and the filtered video are available. The original noise-free video is in general not available. Many *objective* and *hybrid* performance measures, however, use the original video in the calculation of the quality figure. To

overcome this problem, noise is added (Figure 4.2) to recorded noise-free video sequences (F) and these noisy sequences (G) are subsequently filtered by a certain filter, resulting in noise-filtered video (G_F). In this way, all three variants of a video sequence are present for the performance evaluation.

In this chapter, several objective, subjective and hybrid quality measures are introduced for the evaluation of video noise reduction algorithms. The selection is based on a combination of commonly known quality measures and some alternatives that are expected to correlate better with subjective quality.

3.1 Objective quality measures

3.1.1 Signal-to-Noise Ratio and Signal-to-Noise Ratio Improvement

The Signal-to-Noise Ratio (SNR) [18] is one of the most commonly used noise figures in different engineering disciplines. It gives the relation of signal strength to noise strength. The more noise is present compared to the signal strength, the lower the ratio is. Preferably, SNR should be as high as possible. SNR of a filtered video sequence is defined as:

$$SNR(n) = 10 \log_{10} \left(\frac{\sum_{\vec{x}} F(\vec{x}, n)^2}{\sum_{\vec{x}} (F(\vec{x}, n) - G_F(\vec{x}, n))^2} \right) \tag{3.1}$$

where n is a given image.
The SNR Improvement ($SNRI$) [18] gives the improvement in SNR by the noise filtering of the image. The definition is given as:

$$SNRI(n) = 10 \log_{10} \left(\frac{\sum_{\vec{x}} (F(\vec{x}, n) - G(\vec{x}, n))^2}{\sum_{\vec{x}} (F(\vec{x}, n) - G_F(\vec{x}, n))^2} \right) \tag{3.2}$$

where n is a given image.

3.1.2 Peak-Signal-to-Noise Ratio and Peak-Signal-to-Noise Ratio Improvement

The Peak-Signal-to-Noise Ratio ($PSNR$) [18] is related to SNR. Where in SNR the actual signal strength of the video is used, in $PSNR$ the maximum or peak signal strength is used. This measure, therefore, takes into account only the amount of noise and is unrelated to the video content. Because of that reason, $PSNR$ of different video images can be better compared to one another than SNR. $PSNR$ of a filtered video sequence is defined as:

$$PSNR(n) = 10 \log_{10} \left(\frac{N {F_{max}}^2}{\sum_{\vec{x}} (F(\vec{x}, n) - G_F(\vec{x}, n))^2} \right) \tag{3.3}$$

where $F_{max} = 255$ in the case of an 8-bits feature value format, N is the number of pixels per image and n is the image number. $PSNR$ can also be rewritten to

compare the improvement in noise reduction, as *PSNRI*. *PSNRI*[1] is defined as:

$$PSNRI(n) = 10\log_{10}\left(\frac{\sum_{\vec{x}}(F(\vec{x},n)-G(\vec{x},n))^2}{\sum_{\vec{x}}(F(\vec{x},n)-G_F(\vec{x},n))^2}\right) \tag{3.4}$$

where n is the image under consideration.

3.1.3 Mean Distortion or Mean Average Error

Noise reduction algorithms tend to, apart from reducing the noise strength, distort the content of the video as well. This is sometimes visible, for instance as a loss of sharpness or blur. The Mean Distortion (*MD*) [1] aims at measuring this effect. It evaluates the differences between the noise filtered image and the original noise-free image. *MD* should ideally be zero. The *MD* is equal to the Mean Average Error (*MAE*) [18]. *MD* of a filtered video sequence is defined as:

$$MD(n) = \frac{1}{N}\sum_{\vec{x}}|F(\vec{x},n)-G_F(\vec{x},n)| \tag{3.5}$$

where N is the number of pixels per image n.

3.1.4 Mean Square Error

The Mean Square Error (*MSE*) [18] is measuring the same property as *MD*, i.e. the distortion of a noise filtered image with respect to the original one. *MSE* of a filtered video sequence is defined as:

$$MSE(n) = \frac{1}{N}\sum_{\vec{x}}(F(\vec{x},n)-G_F(\vec{x},n))^2 \tag{3.6}$$

where N is the number of pixels in the image n.
Like *MD* and *MAE*, *MSE* should be zero in the ideal case. *MSE* can also be used in the calculation of *SNRI*. The following relation exists between *SNRI* and *MSE* [1]:

$$SNRI(n) = 10\log_{10}\left(\frac{MSE_{G(\vec{x})}(n)}{MSE_{G_F(\vec{x})}(n)}\right) \tag{3.7}$$

where n is a given image.

3.1.5 Mean Busyness

The *Busyness* of each pixel in an image is defined as the median of the absolute vertical and horizontal luminance differences in a 3x3 mask, as defined in Equation 3.8.

$$Busyness = \text{median}\{|d_j(\vec{x})|\} \tag{3.8}$$

where $d_j(\vec{x})$ is the jth vertical and horizontal spatial luminance difference in a 3x3 mask, centered around position $\vec{x}$. The median value of the set {a,b,..}

[1] As can be seen, *SNRI* and *PSNRI* have the same definition.

is notated as $median\{a, b, ...\}$ and N represents the number of pixels in the image n.

These *Busyness* figures per pixel are averaged for the entire image and the resulting figure is called the Mean Busyness (MB) [1,18].

$$MB(n) = \frac{1}{N} \sum_{\vec{x}} Busyness \tag{3.9}$$

where the *Busyness* is defined in Equation 3.8. The figure indicates the amount of spatial variation in an image and can be compared with the spatial variation in other images, that are for instance filtered by another noise reduction algorithm. The lower the MB, the larger the smoothing of the noise reduction algorithm.

3.1.6 Stability

When images are impaired by different noise sources with the same characteristics (noise power, noise bandwidth), noise reduction algorithms can still have different output images. This effect is undesired, because noise reduction should be independent of the way noise with the same characteristics impairs video. The Stability measure (ST) [1,18] gives the average consistency for images filtered after being degraded with different noise sources that have the same noise strength and type. ST is defined as:

$$ST(n) = \frac{1}{N} \frac{1}{M} \sum_{j} \sum_{\vec{x}} \{I_{Aj}(\vec{x})\} \tag{3.10}$$

where M is the number of pairs of images processed with different noise sources, n is the image under consideration, N is the number of pixels per image and $I_{Aj}(\vec{x})$ is an indicator function defined as:

$$I_{Aj}(\vec{x}) = \begin{cases} 1 & , (g_1{}^j(\vec{x}) = g_2{}^j(\vec{x})) \\ 0 & , (\text{otherwise}) \end{cases} \tag{3.11}$$

where $g_1{}^j(\vec{x})$ and $g_2{}^j(\vec{x})$ is a pair of consecutively filtered images, impaired by noise source 1 and noise source 2, respectively.

3.1.7 Correct Processing Ratio

A noisy image consists in general of two different types of pixels, i.e. the noise degraded pixels and the pixels that have not been degraded by noise (original image pixels). It is desirable that the video noise reduction algorithm modifies only the pixels that are degraded by the noise, while it does not modify the original image pixels. Knowing this, four kinds of filter actions can be distinguished [1]:

1. Pixels that have been affected by the noise are modified[2].

[2]It is assumed in the measure that a modification of a noisy pixel always result in $|F(\vec{x}) - G_F(\vec{x})|_{\vec{x}_i} < |F(\vec{x}) - G(\vec{x})|_{\vec{x}_i}$ for $i = 0, 1, ...N$.

2. Pixels that have not been affected by the noise, i.e. original image pixels, are modified.
3. Pixels that have been affected by the noise are not modified.
4. Pixels that have not been affected by the noise are not modified.

Of these filter actions, the first and the fourth ones are correct behaviour of the algorithm, while the second and the third ones are incorrect behaviour of the algorithm.
The mean ratio of correct processed pixels to all processed pixels (both correct and incorrect processed ones) is called the Correct Processing Ratio (*CPR*) [1,18], which has the following definition:

$$CPR(n) = \frac{1}{N} \sum_{\vec{x}} \{I_B(\vec{x}) + I_C(\vec{x})\} \tag{3.12}$$

where N is the number of pixels per image n and $I_B(\vec{x})$ and $I_C(\vec{x})$ are indicator functions defined as:

$$I_B(\vec{x}) = \begin{cases} 1 & ,(G(\vec{x},n) = F(\vec{x},n) \wedge G_F(\vec{x},n) = G(\vec{x},n)) \\ 0 & ,(\text{otherwise}) \end{cases} \tag{3.13}$$

$$I_C(\vec{x}) = \begin{cases} 1 & ,(G(\vec{x},n) \neq F(\vec{x},n) \wedge G_F(\vec{x},n) \neq G(\vec{x},n)) \\ 0 & ,(\text{otherwise}) \end{cases} \tag{3.14}$$

3.1.8 Flat Field Noise Smoothing Ability

The Flat Field Noise Smoothing Ability (*FFNSA*) [19] is a measure for the maximum reachable noise removal capability. In the flat fields, which are the parts of an image with no detail and no motion neighbourhood adaptivity, the maximum noise removal of the noise reduction algorithm can be reached. The *FFNSA* measure can be implemented using the *SNR* or *PSNR*. A simple approximating measurement of the *FFNSA* can be done by measuring the noise reduction for a sequence with a constant luminance value that is degraded with noise resulting, for instance, in a 26 or 32 dB *PSNR* sequence and subsequently filtered by different noise reduction algorithms. The definition for *FFNSA* is:

$$FFNSA_{PSNRI(n)} = 10 \log_{10} \left(\frac{\sum_{\vec{x}} (F(\vec{x},n) - G_N(\vec{x},n))^2}{\sum_{\vec{x}} (F(\vec{x},n) - G_F(\vec{x},n))^2} \right)_{flat\ fields} \tag{3.15}$$

where n is the image under consideration, N is the number of pixels per image and *flat fields* are the regions of the image with little or no high-frequency spatial content.

3.1.9 Detail-Preservation Capability

The Detail-Preservation Capability (*DPC*) [19] measures the amount of image detail that is retained after filtering. Noise filters, in general, have a smoothing or low-pass character [7]. This means that details like edges and textures are

likely to be degraded by the filtering process. *DPC* can be implemented by using distortion measures, such as *PSNR*, *MAE* or *MD*. Because detail is mainly made up of high frequency image content, the following definition for *DPC* can be given, here based on *PSNR*[3]:

$$DPC_{PSNR(n)}(n) = 10\log_{10}\left(\frac{NF_{max}{}^2}{\sum_{\vec{x}}(F(\vec{x},n) - G_F(\vec{x},n))^2}\right)_{high-pass} \tag{3.16}$$

where F_{max} is 255 for an 8-bit feature value format, N is the number of pixels per image and n is the image number. The cut-off frequency for the *high-pass* operation can be deducted from several weighting curves [12–14] described in the last part of this chapter (Section 3.3) to calculate weighted *SNR* and *PSNR* measures.

Detail-preservation should ideally be as large as possible. This means that *DPC* should be ideally infinite, if based on the *PSNR* evaluation measure.

3.2 Subjective quality measures

The subjective quality can be measured directly with the help of observers in a viewing experiment with still images or a video sequence. When taking into account the three available video categories for the evaluation, namely noise-free original video F, noise-impaired video G and noise-filtered video G_F, there are three approaches to evaluate the quality of the noise reduction algorithms:

- **Positive approach.** Assess the quality improvement of filtered noisy video sequences G_F compared to the noise-impaired video G.
- **Negative approach.** Assess the impairments of filtered noisy video sequences G_F compared to the noise-free original video F.
- **Multiple comparison.** Assess the quality of filtered noisy video sequences G_F resulting from different noise reduction algorithms. Here a ranking of observed quality of the noise algorithms can be obtained.

Different scales to measure subjective quality have been proposed by several research institutions, notably the European Broadcasting Union (EBU), International Telecommunications Union (ITU) and the British Telecom Research Laboratories. Apart from the type of rating scale a scale can be chosen to be continuous or discrete. The latter scale type is also called a grading scale. With the continuous scale the total range of the scale can be used to vote, whereas with the grading scale only the mentioned grades can be used. The continuous scale has the advantage that observer errors due to rounding of voting scores are avoided. There are basically three types of scales [2,20]:

1. **Quality scale** (Figure 3.1). This continuous scale describes directly the quality of a certain video sequence. Data can be obtained for the perceived

[3]Similar definitions for *MAE*, *MD*, *MSE* and *SNR* can be given.

quality of noisy video sequences or for noise-filtered video sequences or for both at the same time. ITU recommends the use of the double stimulus continuous quality scale method, in which an image or video sequence is compared to a reference image or video sequence, and both are scored on image quality. The images are shown one after the other on a single display with a short time interval between the test and the reference image. The reference and the test image are each shown during 10 seconds, with a three seconds interval. Then, there is a period of 10 seconds for the subjects to make their quality decision (voting).

2. **Impairment scale** (Table 3.1). This grading scale describes the amount of apparent degradation in a video sequence. It can be used to measure the amount of perceived annoyance of certain noise strengths or the annoyance of artefacts in noise-filtered video sequences. ITU recommends the use of the double stimulus impairment scale method (also called the EBU-method) in which an image is compared with an unimpaired reference image. The same experiment timing as described for the double stimulus continuous quality scale method is being used.

3. **Paired comparison method**. This grading method is used to compare the difference in quality between all possible conditions of a processed video sequence, such as e.g. the mutual comparison of different noise reduction algorithms. ITU has paired comparison methods under study, and currently advises the scale as shown in Table 3.2.

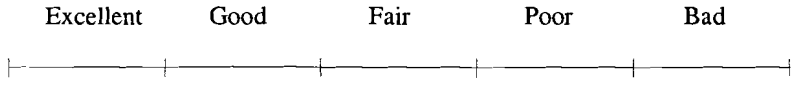

Figure 3.1: *Quality scale recommended by ITU*

3.3 Hybrid quality measures

3.3.1 Weighted *SNR* and Weighted *PSNR*

From several studies about the objectionability of noise for human perception (see for instance [12], it has been found that low noise frequencies are far more annoying than high noise frequencies in luminance values. Several attempts have been made to describe the relationship between the noise frequency and

Table 3.1: *Impairment scale recommended by ITU*

5	Imperceptible
4	Perceptible, but not annoying
3	Slightly annoying
2	Annoying
1	Very annoying

Table 3.2: *Comparison scale recommended by ITU*

+3	Much better
+2	Better
+1	Slightly better
0	The same
-1	Slightly worse
-2	Worse
-3	Much worse

human perception of noise [12–14]. It has been found that subjectively weighted *SNR* and *PSNR* can give a better representation of the subjective quality of noise reduction algorithms than unweighted *SNR* and *PSNR*.
The general relationship between the normal noise power spectrum $N_0(f_x, f_y)$, where f_x and f_y are the frequencies in the horizontal and vertical direction, respectively, and the weighted noise power N_W is:

$$N_W = 10\log_{10}\left(\frac{1}{N}\int_0^{\frac{1}{2}f_{sx}}\int_0^{\frac{1}{2}f_{sy}}|W(f_x,f_y)|^2 N_0(f_x,f_y)df_x df_y\right) \tag{3.17}$$

where $W(f_x, f_y)$ is the weighting function, $N_0(f_x, f_y)$ is the normal noise power and f_s is the sampling frequency of the video signal. The weighting factor is defined as [1]:

$$W(f_x,f_y) = \frac{1}{\left(1+\left(\frac{f_x}{f_0x}\right)\right)\left(1+\left(\frac{f_y}{f_0y}\right)\right)} \tag{3.18}$$

where f_0x is 0.5 MHz and f_0y is 20 cph. The normalising constant N is defined as

$$N = \int_0^{\frac{1}{2}f_{sx}}\int_0^{\frac{1}{2}f_{sy}}|W(f_x,f_y)|^2 df_x df_y \tag{3.19}$$

In [12–14] alternative weighting functions $W(f_x, f_y)$ are given. These measures are: Fujio *SNR/PSNR*, CCIR 4211 *SNR/PSNR* and CCIR 5672 *SNR/PSNR*.

3.3.2 Spatial Distortion Measure

The Spatial Distortion Measure (*SDM*) [16, 17] is a spatial measure that indicates the amount of high frequencies that is affected.

First, the Sobel[4] filtering operation, S, is applied to D_n, that refers to the nth image of the degraded video sequence and O_n which refers to the nth image of the original video sequence. From the result after this filtering, the standard deviation over all pixel values, std_{space}, is taken. Then, the Root Mean Square, RMS_{time}, is taken over time, which is used to combine the n values (for the n images) to one value. The constant C serves to normalise the variance of this measurement.
The definition of this measure is as follows:

$$SDM = RMS_{time}\left(C\left|\frac{std_{space}(S(O_n)) - std_{space}(S(D_n))}{std_{space}(S(O_n))}\right|\right) \tag{3.20}$$

3.3.3 Time-Distortion Measure

The Time-Distortion Measure (TDM) [16,17] indicates how the video sequence is distorted in time and is, therefore, a measure for the amount of distortion in motion. TDM is a time-averaged measure. To compute the TDM, first, the frame difference of the current frame with the previous frame is calculated for the original video, i.e. $\triangle O = O_n - O_{n-1}$ and the same difference is determined for the distorted video, i.e. $\triangle D = D_n - D_{n-1}$ both on pixel basis for the whole frame. Secondly, the Root Mean Square, RMS, is taken of the two calculated differences. Finally, the resulting values are high-pass filtered with a filter with coefficients (-1,2,-1) and the standard deviation of this sequence is the resulting TDM measure. These last operations are summarised in Equation 3.21 as $f_{time}(\ .\ )$, which means $std_{time}(filter(\ .\ ))$. As a consequence, the Time-Distortion Measure (TDM) is defined as:

$$TDM = f_{time}\left(C\ (RMS(\triangle O) - RMS(\triangle D))\right) \tag{3.21}$$

3.3.4 Maximum Distortion

The Maximum Distortion ($Mdis$) [16,17] of the video sequence is an important measure for perceived quality because the human perception is believed to work on a worst case basis (at least for short video sequences).
To compute the $Mdis$, we again calculate $\triangle O = O_n - O_{n-1}$ and $\triangle D = D_n - D_{n-1}$ as for TDM. Then, the standard deviation over pixel values, std_{space}, is taken for $\triangle O$ and $\triangle D$. The log_{10} and the maximum value are then taken over the quotient of these values. The result is normalised with the constant C.
This leads to the definition for $Mdis$ as:

$$Mdis = max_n\left(C\log_{10}\left(\frac{std_{space}(\triangle D)}{std_{space}(\triangle O)}\right)\right) \tag{3.22}$$

[4]Sobel filtering is a well known high-pass filtering method that passes spatial edges of video.

3.4 Conclusions

Several objective, hybrid and subjective evaluation measures have been explained in detail. These measures will be used in Chapter 5 for the objective, hybrid and subjective evaluation of seven noise reduction algorithms.

Chapter 4

Experimental set-up

During two weeks two experiments were conducted to measure the subjective quality of three different sequences degraded with two grades of noise and filtered with seven different noise reduction algorithms. Each experiment was divided into two sessions of about 30-40 minutes. During these sessions, the subjects was asked to score the overall quality of the sequences and the amount of noise that was visible to them, relative to a reference video sequence. In this chapter, the experimental set-up of the two experiments will be described in detail.

4.1 Goal of the experiments

The goal of the panel test was to compare the subjective quality of seven noise reduction algorithms. The subjective quality evaluation results were, subsequently, used to find out if there was a correlation between objective quality measures and the perceived quality of noise reduced video, and if so, what this correlation was. The results of the two experiments, and their relation with objective and hybrid quality measures will be given in Chapter 5.

4.2 Experimental set-up

4.2.1 Noise reduction algorithms

Seven algorithms were evaluated in the panel test. The algorithms used have been described in detail in Chapter 2 and are summarised in Table 4.1.
All algorithms have been used as specified by the designers in the corresponding articles where the algorithms have been described (See Chapter 2.). The Fuzzy (Subsection 2.8.7) and Perception adaptive algorithms (Subsection 2.8.6) additionally have motion compensation with sub-pixel accuracy in both horizontal and vertical direction. The DNR_mc and Falcon_mc_hp algorithms have additionally sub-pixel accuracy for the (standard) horizontal motion compensation. All the noise settings of the algorithms have been determined after intensive simulations on the same monitor as was used in the panel test and also on another monitor. The settings were found by trying different settings and judging

Table 4.1: *Noise reduction algorithms used in the experiments*[1]

Algorithm	Type	Specifics	Subsection:
Falcon_mc_hp	temporal	HP + hor. MC	2.8.5
Siemens (spat+temp)	spatio-temporal	MC	2.8.1+2.8.6
SWAN + DNR	spatio-temporal	HP + MC	2.8.2+2.8.5
Median + Fuzzy	spatio-temporal	MC	2.8.8+2.8.7
SWAN	spatial	HP	2.8.2
DNR	temporal	-	2.8.5
DNR_mc	temporal	horizontal MC	2.8.5

the resulting filtered sequences on possible artefacts, such as blurring and distortions. Settings were adjusted so that maximum noise removal and minimal distortion of the image content occurred. An expert viewer checked almost all noise settings. The settings can be found in Tables D.1 and D.2 of Appendix D.

4.2.2 Selected sequences

The three test-sequences that were selected for the panel test, were: *Renata*, *Car & Gate* and *Teeny*. One image of the test sequences is shown in Figure 4.1.

Figure 4.1: *The selected sequences for the panel test are, on the left: Renata, on the right: Car & Gate and under: Teeny*

These sequences have been chosen, because they were used in [7] to evaluate the performance of the SWAN algorithm. It was assumed that these test sequences offer a good variety of critical sequences to test the performance of the noise reduction algorithms. Of the three sequences only a selection of the total frames could be shown, due to hard disk limitations (1.5 Gb) of the video simulator. The frames that were used for the experiment are shown in Table 4.2. Also, only a part of the full screen could be shown because of the use of the split-screen. The part of the sequence that was most interesting for testing the quality and the noise removal ability of the noise reduction algorithms has been selected. In *Renata*, the moving woman and the almost still tapestry with much detail were the most important. In *Car & Gate*, the moving car and the background bush with much detail were selected. In *Teeny*, the face and the hair were the parts that were considered to be the most important.

4.2.3 Preparation of the noise-files

The selected sequences were first PAL-encoded, then degraded with white Gaussian noise, low-pass filtered with a cut-off frequency of 5MHz and PAL-decoded again. The procedure is described in Figure 4.2. Two different noise strengths have been used, resulting in sequences with an *PSNR* of 26 dB and 32 dB (both unweighted *PSNR*). Figure 4.2 also indicates how an original noise-free sequence

Table 4.2: *Frames of the sequences used in the experiment*

Sequence	Frames
Renata	10-60
Car & Gate	10-29
Teeny	25-56

can be generated to calculate the amount of noise present in the (un)filtered sequences. Each of the noisy sequences G have been processed with one of the filtering algorithms of Table 4.1, yielding the noise-filtered sequences G_F. The latter were used as stimuli in the panel test.

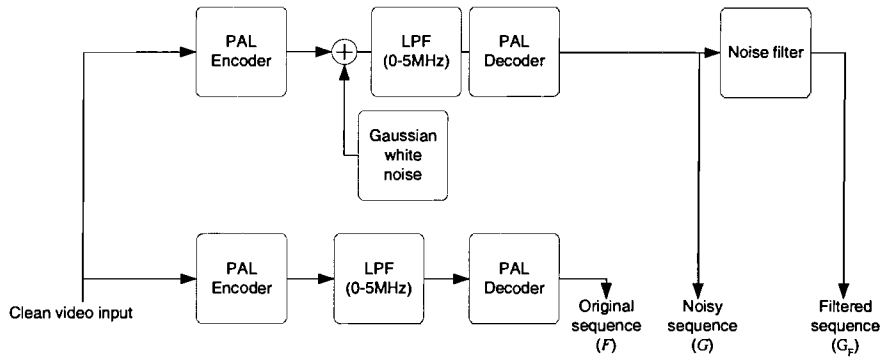

Figure 4.2: *Procedure to make noise files and to prepare noise-free sequences*

4.2.4 Viewing conditions

The tests were conducted in a special viewing room at the Philips Research Laboratories, Eindhoven, The Netherlands. The background illumination and overall lighting was at a rather dimmed setting, that had the same fixed value for every subject. The background illumination[2] was measured to be 13.1 cd/m^2. The viewing distance was 6 times the height of the screen, or about 1.8m. The subjects could sit in a comfortable chair.

The panel test was conducted in a special viewing room under controlled illumination conditions that comply to a large extent to the ITU recommended viewing conditions [20]. The split-screen of the display that was used in the panel test is shown in Figure 4.3.

[2]Background colour temperature was not considered important in the experiment [2].

Table 4.3: *Details of the test monitor*

Brand	Sony
Type	BVM-D24E1WE
Serial no.	2000015
Screen height	0.30 m
Screen	625 lines, 50 Hz

Figure 4.3: *The test monitor set-up of the subjective experiment*

The test monitor was a special Sony Multiformat test monitor, that could display different formats of interlaced video. The shown sequences were all 625 lines/50 Hz interlaced video. The details are given in Table 4.3. The monitor has been adjusted using luminance and contrast controls. The luminance control has been used to adjust the peaking luminance to a value of 180 cd/m^2. The contrast control has been used to adjust the black level with the help of a PLUGE- variant test image. The black level (in a viewing room with no illumination) was 0.2 cd/m^2. The monitor has been tested for homogeneity of the luminance and white-level over the CRT-screen. To this purpose, nine test images have been made all with one small white square at nine different positions and black everywhere else (Figure 4.4.). The luminance level and colour temperature at the positions of the white squares have been measured. The results are given in Table 4.4. It can be seen that the measurement values differ considerably in vertical direction, but only slightly in horizontal direction. Whether the sequence is shown left or right on the CRT-screen makes no difference in the way the sequence is displayed.

Table 4.4: *Luminance and white-level measurement results of monitor*

Square	Luminance $[cd/m^2]$	Colour temperature [K]
1.	313	6048
2.	326	6368
3.	282	6735
4.	316	6457
5.	309	6515
6.	267	6714
7.	308	6341
8.	305	6509
9.	268	6710

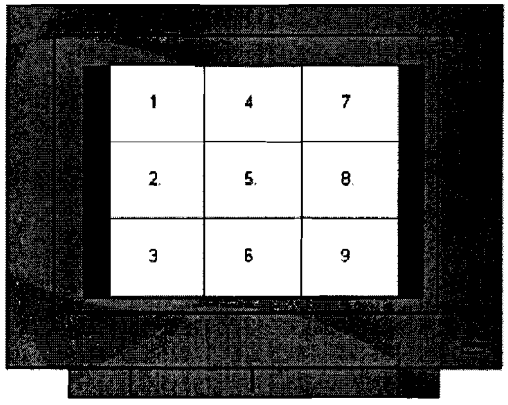

Figure 4.4: *Measurement of luminance and white-level divided over the display*

4.3 Protocol

For the panel test seven algorithms were used in combination with three different sequences and two types of noise. The intention was to let the subject compare filtered sequences generated by different noise reduction algorithms with each other on the split screen. In the first test, five different algorithms were used, of which four algorithms were tested on the two noise degradation levels. The SWAN algorithm was tested only with 26 dB unweighted *PSNR* sequences. The other noise reduction algorithms used in the first experiment were: Falcon_hp_mc, Siemens, SWAN_DNR and Median_Fuzzy.
To compare the SWAN algorithm with the four other algorithms with three sequences on one noise strength, necessitates 4×3 equals 12 stimuli. The comparison of four algorithms with three sequences and two noise strengths leads to (3+2+1)×3×2 equals 36 stimuli. This leads to a total of 48 stimuli for

experiment 1. The accuracy of the experiment has been improved by repeating the stimuli, leading to a total of 96 stimuli. The experiment has been divided into two sessions to obtain a reasonable amount of stimuli per session that could be conducted within 30-40 minutes. During the sessions half of all stimuli were tested and the same stimuli were repeated. In the repetition, the left-right position was switched to average out left- or right-preferences of the subjects. In the second test, five algorithms were used, namely Falcon_hp_mc, DNR, SWAN_DNR, DNR_mc and SWAN. The first four were tested with both noise levels, while the SWAN algorithm was only tested with 32 dB unweighted *PSNR* sequences in the second experiment. This leads to the same amount of stimuli, a total of 96 stimuli with repetitions, that has been divided into two sessions. The algorithms were split in such a way that in experiment 1 complete noise reduction systems were compared, while in experiment 2 different variants of the DNR algorithm were used. Apart from the different noise reduction algorithms and the fact that different subjects participated in the two experiments, each experiment consisted of the same two functional parts: the noise and quality evaluation and the identification of the important parts of the video sequence for the judgement of the amount of noise and quality evaluation.

4.3.1 Experiment

Each experiment contained two separate sessions that were mostly conducted on different days. During the two sessions, all subjects was asked one time to do a Snellen eye-test. Each subjects got a verbal instruction at the start of the first session. In addition, six training sequences were shown to train the subject on the use of the scale. The training sequences contained the believed extreme values on the quality and noise scale. After the training part, the stimuli were shown on the split screen in a different random series per subject. Each stimulus was displayed as long as the subject needed to make his or her evaluation of the stimulus. The subject had to make an evaluation of the right part of the split-screen (Figure 4.3.) with respect to the left part of the screen, showing stimuli generated by different noise reduction algorithms. During the first experiment, a blue screen with an average luminance level was shown after each stimulus. During the second experiment, no blue screen was shown, because in the first experiment some observers complained about it. Before every second session the verbal instruction was repeated. The subject was asked during the second session to fill in test form B.
Each of the two experiments was conducted using the same procedure.

4.3.2 Subjects

For the first experiment 17 subjects have entered the experiment, while in the second experiment 15 subjects participated. In the first experiment, 8 subjects had expertise on video processing, the rest (9) had no expertise on video processing. In the second experiment, there were 4 viewers with video processing experience and the rest (11) had no experience in this area. This information is not used in the statistical analysis.

4.3.3 Test form

Two test forms were used in the two experiments: test forms A and B. The first test form (Figure 4.5.) was used for scoring the questions about the quality and noise perception. On the second test form (Figure 4.6.), the questions about the specific part of the sequences that were considered to be the most important for the noise and quality evaluation could be described.

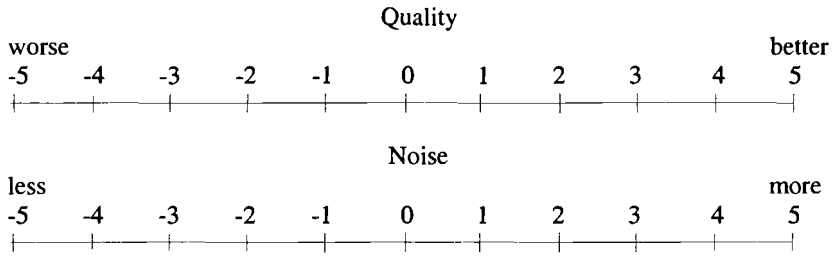

Figure 4.5: *Sample of questions on test form A*

We want to get more information on how you judge each of the sequences. Therefore, we would like to ask you to fill in the following table:

What part of each scene did you use for your judgement?

sequence	Quality	Noise
Renata		
Car & Gate		
Teeny		

Figure 4.6: *Contents of test form B*

4.4 Analysis

All scores of the two experiments have been used as data input in the statistical analysis package SPSS, version 10.0. The scores of the subjects have first been averaged over the two repetitive stimuli. Furthermore, these averaged scores have been normalised per subject by using Equation 4.1.

$$av.\ score_{norm.} = \left(\frac{score - av.\ score}{std.dev.per subject} \right) \tag{4.1}$$

Before input to SPSS, first, the data had to be changed from differential values (scores of the difference between the reference and the stimulus) to scores per stimulus (See Appendix E.). The data analysis with SPSS itself is described in more detail in Chapter 5. The scores have been analysed per noise strength and, also a combined analysis by using both noise strengths has been done. The interaction between subject and the quality and the noise scores has not been taken into account. The analysis in SPSS was an ANOVA on both quality and noise as dependent variables, noise reduction algorithms and test sequences as fixed factors and noise strength as mentioned earlier (either 26 dB, 32 dB or both 26 and 32 dB *PSNR*). As a post-hoc analysis, Tukey's algorithm was used.

4.5 Conclusions

In this chapter, the goal of the conducted experiments has been described. Also a description of the experimental set-up is given with an elaboration on the used noise reduction algorithms, selected sequences, used preparation method for the noise-files and viewing condtions. After that, the protocol is given in detail. Finally, the analysis method of the results is explained.

Chapter 5

Evaluation results

Seven noise reduction algorithms have been elaborately described in Chapter 2. These algorithms have been evaluated using the objective evaluations measures described earlier in Chapter 3. The objective evaluation measures that are used for the evaluation are: *SNR*, *PSNR*, *SNRI*, *PNRI*, *MD*, *MSE* and *MB*, *ST*, *CPR*, *FFNSA* and *DPC*. The hybrid evaluation measures that are used are: Fujio *SNR* and *PSNR* [12], CCIR 4211 *SNR* and *PSNR* [13], CCIR 5672[1] *SNR* and *PSNR* [14], *SDM*, *TDM* and *Mdis*. In this chapter, first, an overview of the ranking results of the objective and hybrid evaluation measures are presented (Section 5.1). Secondly, the subjective test results are given in Section 5.2.

5.1 Objective and hybrid evaluation results

All objective evaluation results are given in Appendix A. The hybrid evaluation measures are given in Appendix B. In this section, only an overview of these results will be given.
In Table 5.1 and Table 5.2 are all objective and hybrid evaluation measure results summarised. The tables give the rank number of the noise reduction algorithm obtained by ranking the averaged evaluation measure results for the three test sequences *Renata*, *Car & Gate* and *Teeny*. The noise reduction algorithms have been ordered from left to right by overall rank number. From this tables, it is easier to make a comparison of the consistency of the different evaluation measures. The *SNR*, *PSNR*, *SNRI*, *MD*, *MSE* and *MB* evaluation measures have the same results for the 26 dB *PSNR* sequences. Also for 32 dB *PSNR* sequences have the mentioned evaluation measures a similar behaviour, although this similarity shows some small deviations for the *MSE* evaluation measure, while the ranking of the noise reduction algorithms with the *MB* evaluation measure deviates heavily from the others. For the *MSE* evaluation measure the rankings of SWAN and DNR_mc have been switched as compared to the ranking of the other members of the earlier mentioned group of consistent evaluation measures.

[1] *SNR* and *PSNR* in combination with Fujio, CCIR 4211 and CCIR 5672 weighting will be called, respectively, *WSNR1* and *WPSNR1*, *WSNR2* and *WPSNR2* and *WSNR3* and *WPSNR3*.

Table 5.1: *Ranking of noise reduction algorithms by performance measured by all noise reduction evaluation measures 26 dB PSNR sequences*

	SWANDNR	Siemens	DNRmc	SWAN	Med.Fuzzy	DNR	Falconmchp
SNR	1	2	3	4	5	6	7
PSNR	1	2	3	4	5	6	7
(P)SNRI	1	2	3	4	5	6	7
MD	1	2	3	4	5	6	7
MSE	1	2	3	4	5	6	7
MB	1	2	3	4	5	6	7
ST	2	1	6	4	5	3	7
CPR	3	4	1	5	7	6	2
FFNSA	2	3	1	7	6	4	5
DPC	1	2	3	4	6	5	7
WSNR1	1	4	3	2	5	6	7
WPSNR1	1	4	3	2	5	6	7
WSNR2	1	4	3	2	6	5	7
WPSNR2	1	4	3	2	5	6	7
WSNR3	1	4	3	2	6	5	7
WPSNR3	1	4	3	2	5	6	7
SDM	1	3	4	2	5	6	7
TDM	4	1	5	6	3	7	2
Mdis	3	2	1	5	7	6	7

Another group of evaluation measures that show a highly consistent behaviour is the group of the weighted *SNR* and *PSNR* evaluation measures. All of these measures show a highly consistent ranking of the algorithms for 26 dB *PSNR* sequences. A small difference exists the ranking of the Median_Fuzzy and the DNR noise reduction algorithm with *WSNR2* and *WSNR3* evaluation measures. These rankings have been switched as compared with the other weighted *SNR* and *PSNR* measures. For the 32 dB *PSNR* sequences, a highly consistent ranking results with the weighted *SNR* and *PSNR* evaluation measures. Only the *WSNR1* and *WPSNR1* evaluation measures show a different ranking for the Median_Fuzzy, DNR and Siemens noise reduction algorithms, as compared to the other weighted *SNR* and *PSNR* evaluation measures.

5.2 Subjective evaluation results

5.2.1 Analysis of panel test

The raw scores obtained from the subjects have been manipulated using the procedures described in Chapter 4 and Appendix E. The resulting data have been analysed with the statistical package SPSS, version 10.0. The results will be described in the following paragraphs.

Table 5.2: *Ranking of noise reduction algorithms by performance measured by all noise reduction evaluation measures 32 dB PSNR sequences*

	SWANDNR	SWAN	DNRmc	Med.Fuzzy	DNR	Siemens	Falconmchp
SNR	1	5	3	2	4	6	7
PSNR	1	5	3	2	4	6	7
(P)SNRI	1	5	3	2	4	6	7
MD	1	5	3	2	4	6	7
MSE	1	3	5	2	4	6	7
MB	4	5	2	3	7	1	7
ST	3	7	5	4	2	1	6
CPR	4	5	3	6	7	2	1
FFNSA	1	5	4	2	7	6	3
DPC	2	6	3	4	5	1	7
WSNR1	1	2	3	5	6	4	7
WPSNR1	1	2	3	5	6	4	7
WSNR2	1	2	5	4	3	6	7
WPSNR2	1	2	5	4	3	6	7
WSNR3	1	2	5	4	3	6	7
WPSNR3	1	2	5	4	3	6	7
SDM	3	4	2	5	6	1	7
TDM	5	6	3	4	7	1	2
Mdis	4	6	5	3	1	2	7

5.2.2 Experiment 1

A two-tailed Pearson's bivariate correlation analysis has been conducted on the Quality and Noise scores. The correlation between the Quality and Noise scores was -0.665 with significance level < 0.01. From this analysis, it can be concluded that Quality and Noise are not highly correlated. Therefore, the results for both Quality score and the results for the Noise score will be described separately.

An ANOVA on the data of experiment 1 has been conducted with Quality as dependent variable and Sequence, Algorithm and Noise_type as fixed variables. The outcome showed that there is an effect of the variable Algorithm on Quality at significance level < 0.001 ($F = 183.861$, $df = 4$). There is an interaction between the variable Sequence and the variable Algorithm at significance level < 0.001 ($F = 9.371$, $df = 8$). Furthermore, the variable Noise_type has an effect on Quality at significance level $= 0.009$ ($F = 6.804$, $df = 1$).
The ranking of the algorithms used in experiment 1 on basis of the Quality scores using Tukey's algorithm is shown in Figure 5.1. In this figure the underlining of the SWAN_DNR and Falcon_mc_hp algorithms means that these algorithms could not be distinguished by Tukey's algorithm and these algorithms score comparably good.

Siemens > SWAN_DNR > Falcon_mc_hp > SWAN > Median_Fuzzy

Figure 5.1: *Ranking of the five noise reduction algorithms on basis of the Quality score*

The Siemens algorithm scores the best, next best are the SWAN_DNR and Falcon_mc_hp algorithms that are undistinguishable from each other, followed by the SWAN algorithm and the worst is the Median_Fuzzy noise reduction algorithm. In Figure 5.2 the five algorithms of the first experiment are ranked by quality perception. The figure gives the combined result for both noise degradations.

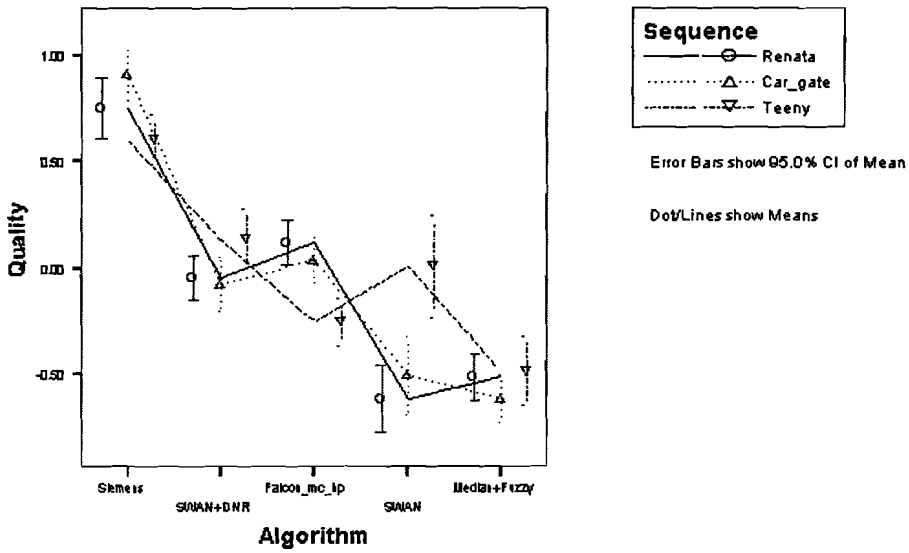

Figure 5.2: *Quality ranking of five noise reduction algorithms of experiment 1*

An ANOVA on the Noise score from experiment 1 has been conducted with Noise as dependent variable and Sequence, Algorithm and Noise_type as fixed variables. The results showed that there is an effect of the Algorithm variable on Noise at significance level < 0.001 ($F = 59.871$, $df = 4$). Furthermore, there is an interaction between the Sequence and the Algorithm variable at significance level < 0.001 ($F = 5.324$, $df = 8$). The ranking of the algorithms used in experiment 1 on basis of the Noise scores using Tukey's algorithm is shown in Figure 5.3. There is no meaningful difference between the algorithms that are underlined and no difference between the algorithms that are overlined. The Falcon_mc_hp algorithm belongs to both subsets, distinguished by respectively, underlining and overlining. The Siemens algorithm performs the best, while the Median_Fuzzy algorithm performs the worst.

Siemens > SWAN_DNR > Falcon_mc_hp > SWAN > Median_Fuzzy

Figure 5.3: *Ranking of the five noise reduction algorithms on basis of the Noise score*

In Figure 5.4, the five algorithms of the first experiment are ranked by the Noise scores.

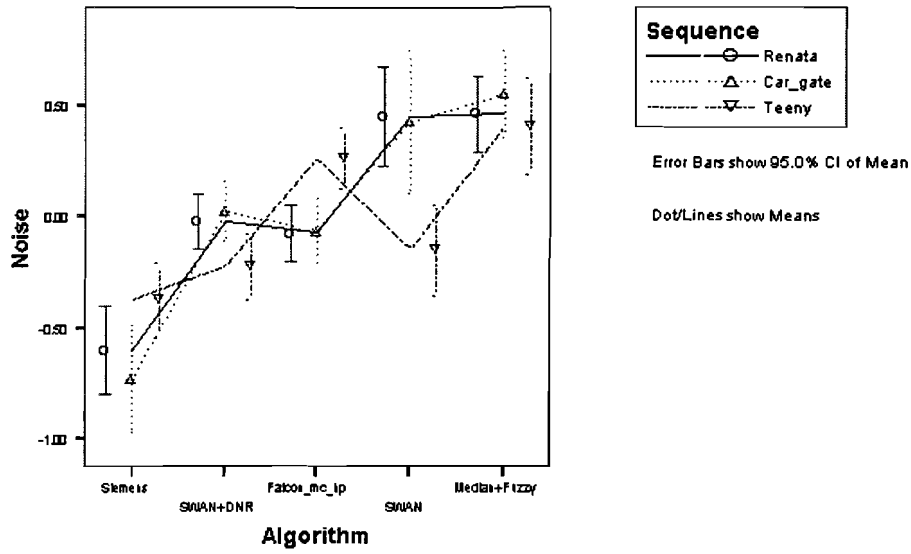

Figure 5.4: *Noise ranking of five noise reduction algorithms of experiment 1*

The Siemens noise reduction algorithm performs for both Quality as Noise scores the best. Next come SWAN_DNR together with Falcon_mc_hp. SWAN could not be distinguished on Noise performance from Falcon_mc_hp. On Quality performance, SWAN was worse than SWAN_DNR and Falcon_mc_hp. The worst performing algorithm for both Quality and Noise scores was the Median_Fuzzy algorithm.

Effect of noise type

In the earlier performed multivariate analysis, it was found that the Noise_type has a significant influence on the Quality score. Therefore, an analysis per Noise_type is necessary.
The Quality ranking of the five algorithms with 26 dB *PSNR* sequences is given in Figure 5.5. The Siemens algorithm performs the best, followed by the SWAN_DNR algorithm, then come Falcon_mc_hp and SWAN that are comparably good. The worst algorithm is Median_Fuzzy.

$$\text{Siemens} > \text{SWAN\_DNR} > \overline{\text{Falcon\_mc\_hp} > \text{SWAN}} > \text{Median\_Fuzzy}$$

Figure 5.5: *Ranking of the five noise reduction algorithms on basis of the Quality score for 26 dB PSNR sequences*

The Noise ranking of the five algorithms is given in Figure 5.6. The result of this ranking is exactly the same as the ranking based on Quality.

Siemens > SWAN_DNR > Falcon_mc_hp > SWAN > Median_Fuzzy

Figure 5.6: *Ranking of the five noise reduction algorithms on basis of the Noise score for 26 dB PSNR sequences*

Therefore, it can be concluded that the Quality and Noise ranking have exactly the same outcome for the 26 dB *PSNR* sequences.
The Quality ranking of the four algorithms with sequences of 32 dB *PSNR* is given in Figure 5.7. The Siemens noise reduction algorithm is the best perceived algorithm, next come the Falcon_mc_hp and the SWAN_DNR algorithms and, finally, the SWAN_DNR and Median_Fuzzy algorithms. The SWAN_DNR algorithm has a combined second and third rank, together with, respectively, Falcon_mc_hp and Median_Fuzzy.

Siemens > Falcon_mc_hp > SWAN_DNR > Median_Fuzzy

Figure 5.7: *Ranking of the four noise reduction algorithms on basis of the Quality score for 32 dB PSNR sequences*

The Noise performance ranking is shown in Figure 5.8. This ranking is the same as the Quality ranking.

Siemens > Falcon_mc_hp > SWAN_DNR > Median_Fuzzy

Figure 5.8: *Ranking of the four noise reduction algorithms on basis of the Noise score with 32 dB PSNR sequences*

Both Quality and Noise performance result in the same ranking for 32 dB *PSNR* sequences.

5.2.3 Experiment 2

As for experiment 1, a two-tailed Pearson's bivariate correlation analysis has been conducted on the Quality and Noise scores for experiment 2. The correlation between the Quality and Noise scores was -0.707 with significance level < 0.01. From this analysis, it can be concluded that Quality and Noise are not highly correlated. Therefore, the results for the Quality score as well as the results for the Noise score will be described.

An ANOVA on the data of experiment 2 has been conducted with Quality as dependent variable and Sequence, Algorithm and Noise_type as fixed variables. The results showed that there is an effect of the variable Algorithm on Quality

at significance level < 0.001 (F = 30.981, df = 4). There is an interaction between the variable Sequence and the variable Algorithm at significance level < 0.001 (F = 6.317, df = 8).
The ranking of the algorithms used in experiment 2 on basis of the Quality scores using Tukey's algorithm is shown in Figure 5.9. In this figure the horizontal line over the SWAN_DNR and Falcon_mc_hp means that these algorithms could not be distinguished by the Tukey's algorithm and these algorithms score comparably good or bad.

SWAN_DNR > DNR_mc > Falcon_mc_hp > SWAN > DNR

Figure 5.9: *Ranking of the five noise reduction algorithms on basis of the Quality score*

The SWAN_DNR, DNR_mc and Falcon_mc_hp algorithms score the best. These algorithms are undistinguishable on basis of the Quality performance. Next comes the SWAN, followed by the DNR noise reduction algorithm that performs the worst on Quality. In Figure 5.10 the five algorithms of the second experiment are ranked by Quality score.

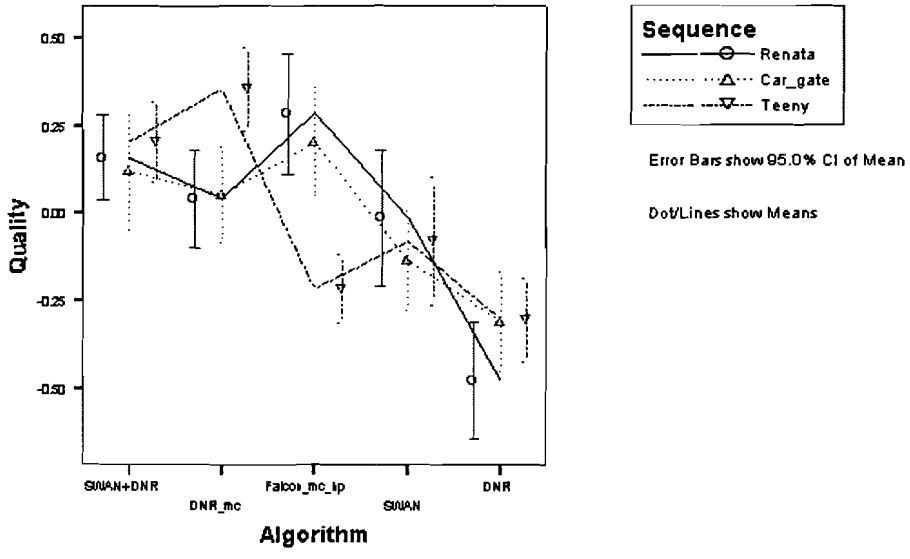

Figure 5.10: *Quality ranking of five noise reduction algorithms of experiment 2*

An ANOVA on the Noise scores of experiment 2 has been conducted with Noise as dependent variable and Sequence, Algorithm and Noise_type as fixed variables. The results showed that there is an effect of the Algorithm variable on Noise at significance level < 0.001 ($F = 29.004$, $df = 4$). Furthermore, there is an interaction between the Sequence and the Algorithm variable at significance level $= 0.002$ ($F = 3.098$, $df = 8$). The ranking of the algorithms used in experiment 1 on basis of the Noise scores using Tukey's algorithm is shown in Figure 5.11. The DNR_mc, SWAN_DNR and Falcon_mc_hp noise reduction algorithms are undistinguishable and the best performing algorithms. The Falcon_mc_hp and the SWAN algorithms are equally good performing and the DNR algorithm is the worst performing.

DNR_mc > SWAN_DNR > Falcon_mc_hp > SWAN > DNR

Figure 5.11: *Ranking of the five noise reduction algorithms on basis of the Noise score*

In Figure 5.12, the five algorithms of the second experiment are ranked by the Noise score.

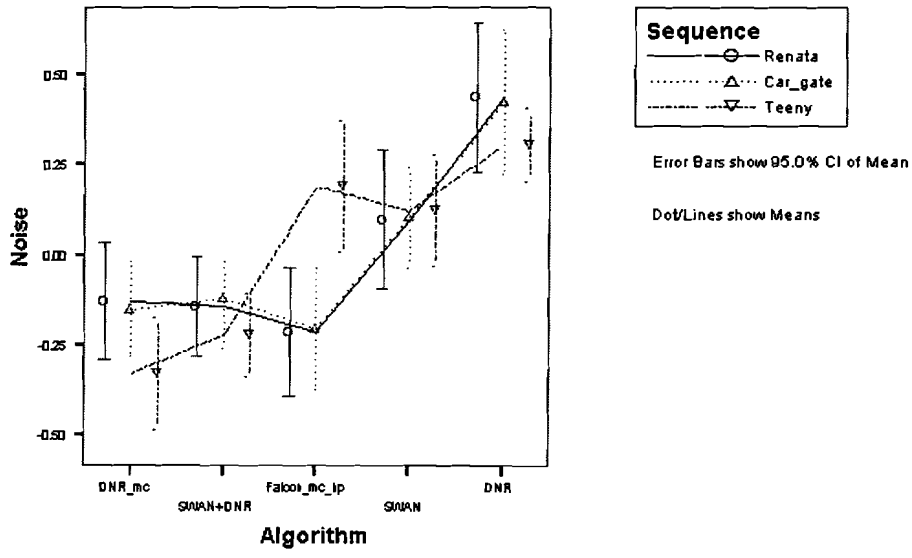

Figure 5.12: *Noise ranking of five noise reduction algorithms of experiment 2*

The performance on Quality and Noise was the same. The only difference was that Falcon_mc_hp and SWAN algorithms could not be distinguished by the Noise scores.

Effect of noise type

Because the multivariate analysis could not give a conclusive judgement about the effect of the variable Noise_type on the Noise and Quality scores, individual results per Noise_type are given in this subsection.
The Quality performance ranking of the four algorithms with 26 dB *PSNR* sequences is shown in Figure 5.13. The SWAN_DNR algorithm performs the best, after this come the Falcon_mc_hp and DNR_mc algorithms. The DNR algorithm performs the worst.

SWAN_DNR > Falcon_mc_hp > DNR_mc > DNR

Figure 5.13: *Ranking of the four noise reduction algorithms on basis of the Quality score with 26 dB PSNR sequences*

The Noise performance ranking is shown in Figure 5.14. The SWAN_DNR algorithm and the DNR_mc algorithm perform the best and are the same performing, next come the DNR_mc and Falcon_mc_hp algorithms that are comparable in performance and last comes the DNR algorithm.

SWAN_DNR > DNR_mc > Falcon_mc_hp > DNR

Figure 5.14: *Ranking of the four noise reduction algorithms on basis of the Noise score with 26 dB PSNR sequences*

On both Noise and Quality scores for 26 dB *PSNR* sequences, SWAN_DNR performs the best (on Noise together with DNR_mc). DNR performs the worst for both scores. DNR_mc and Falcon_mc_hp score comparably for both scores and are better performing than DNR.
The Quality performance ranking with 32 dB *PSNR* sequences is shown in Figure 5.15. The DNR_mc, Falcon_mc_hp and SWAN_DNR algorithms perform the best and are comparably good, next come SWAN_DNR and SWAN that have a similar performance and the worst algorithm is DNR.

DNR_mc > Falcon_mc_hp > SWAN_DNR > SWAN > DNR

Figure 5.15: *Ranking of the five noise reduction algorithms on basis of the Quality score with 32 dB PSNR sequences*

The Noise performance ranking with 32 dB *PSNR* sequences is shown in Figure 5.16. The DNR_mc, SWAN_DNR and Falcon_mc_hp perform the best after

Table 5.3: *Percentage of subjects that use detailed or undetailed parts of the sequences for the judgement of quality*

	Renata	Car& Gate	Teeny
Detailed parts	60%	60%	26%
Undetailed parts	40%	40%	74%

Table 5.4: *Percentage of subjects that use detailed or undetailed parts of the sequences for the judgement of the amount of noise*

	Renata	Car& Gate	Teeny
Detailed parts	50%	45%	10%
Undetailed parts	50%	55%	90%

that come SWAN_DNR, Falcon_mc_hp and SWAN and last is the DNR algorithm.

DNR_mc > SWAN_DNR > Falcon_mc_hp > SWAN > DNR

Figure 5.16: *Ranking of the four noise reduction algorithms on basis of the Noise score with 32 dB PSNR sequences*

For both Quality and Noise scores with 32 dB *PSNR* sequences, DNR_mc, SWAN_DNR, Falcon_mc_hp and SWAN scored better than the DNR noise reduction algorithm. DNR scored the worst for both scores.

5.2.4 Important parts of the sequences used for the Quality and Noise evaluation

Table 5.3 summarises the results of the second part of the subjective test of experiment 1 and 2. The table gives the percentage of subjects that uses the detailed or the undetailed parts of the sequences for the Quality and Noise evaluation in the first part of the test. For the sequences *Renata* and *Car & Gate* no explicit preference can be given for detailed or undetailed parts. For the *Teeny* sequence a clear preference is present for undetailed areas of the sequence. The results of Table 5.4 leads in a similar way to the conclusion that only for the *Teeny* sequence a clear preference for undetailed areas was present.

5.3 Conclusions

Results have been given of the objective and hybrid evaluation measures. The noise algorithms are ranked on the overall performance for 26 and 32 dB *PSNR* sequences. Also an overview has been given of all objective and hybrid evaluation measures (Table 5.1 and Table 5.2). The ranking of the algorithms is for the 26 dB *PSNR*, in order of goodness: SWAN_DNR, Siemens, DNR_mc, SWAN, Median_Fuzzy, DNR and Falcon_mc_hp. For 32 dB *PSNR* sequences, in order of goodness: SWAN_DNR, SWAN, DNR_mc, Median_Fuzzy, DNR, Siemens and Falcon_mc_hp.
From this table it became clear that there was a strong consistency for *SNR*, *PSNR*, *SNRI*, *MD*, *MSE* and *MB* evaluation measures on 26 dB *PSNR* sequences. For 32 dB *PSNR* sequences the same group of evaluation measures was consistent, only small differences arose for the *MSE* evaluation measure, while the ranking of the noise reduction algorithms of the *MB* evaluation measure deviated heavily. Another group of evaluation measures that show a highly consistent behaviour is the group of the weighted *SNR* and *PSNR* evaluation measures. There were only small deviations within this group.
Results for the subjective test show different rankings for the noise reduction algorithms.
Overall in experiment 1, the Siemens algorithm was the best, for both quality as noise performance. Next comes SWAN_DNR together with Falcon_mc_hp. SWAN could not be distinguished on noise performance from Falcon_mc_hp. On quality perfomance, SWAN was worse than SWAN_DNR and Falcon_mc_hp. The worst performing algorithm for both quality and noise performance was the Median_Fuzzy algorithm.
Subsequently, the effect of noise strength on quality and noise performance has been described. This is done by examining the quality and noise performance for 26 dB and for 32 dB *PSNR* sequences individually. For 26 dB *PSNR* sequences, the Siemens algorithm performs the best, followed by the SWAN_DNR algorithm, then come Falcon_mc_hp and SWAN that are comparably good. The worst algorithm is Median_Fuzzy. For 32 dB *PSNR* sequences, the Siemens noise reduction algorithm is the best perceived algorithm, next come the Falcon_mc_hp and the SWAN_DNR algorithms and, finally, the SWAN_DNR and Median_Fuzzy algorithms. The SWAN_DNR algorithm has a combined second and third rank, together with respectively, Falcon_mc_hp and Median_Fuzzy. Results for quality and noise performance for the two different noise strengths are exactly the same.
In experiment 2, the SWAN_DNR, DNR_mc and Falcon_mc_hp algorithms score the best. These algorithms are undistinguishable on basis of the quality performance. Next comes the SWAN, followed by the DNR noise reduction algorithm that performs the worst on quality. The noise performance was the same, only Falcon_mc_hp and SWAN algorithms could not be distinguished from each other. For both noise and quality performance of noise reduction on 26 dB *PSNR* sequences, SWAN_DNR scored the best (for noise together with DNR_mc). DNR performs the worst for both scores. DNR_mc and Falcon_mc_hp score comparably for both quality and noise performance and are

better performing than DNR. For both quality and noise performance on 32 dB *PSNR* sequences, DNR_mc, SWAN_DNR, Falcon_mc_hp and SWAN scored better than the DNR noise reduction algorithm. DNR scored the worst for quality and noise.
The observers were also asked to indicate important areas for judgement of quality and noise. Results showed that for the sequences *Renata* and *Car & Gate* no explicit preference can be given for detailed or undetailed parts. For the *Teeny* sequence a clear preference is present for undetailed areas of the sequence. These results are applicable for quality as well as noise judgement.

Chapter 6

Towards an objective measure for subjective quality

In this chapter, the evaluation data of the objective and hybrid measures will be used to try to find a linear combination of these measures that approximates the data obtained from the subjective test. In this way, the results obtained from the subjective experiment could be repeated by applying a certain weighted combination of objective and hybrid evaluation measures for similar noise sequences.

6.1 Correlation between averaged Quality and Noise scores in the subjective test

Since the effect of the variable Sequence on the Quality and Noise scores was not statistically significant (Section 5.2), the Quality and Noise scores have been averaged over the three different sequences (*Renata*, *Car & Gate* and *Teeny*). These averaged scores have been used as input for a two-tailed Pearson's correlation analysis between Quality and Noise variables. For experiment 1, the outcome of this analysis showed that there was a correlation of -0.988 at a significance level of 0.01. For experiment 2, the correlation was found to have a value of -0.993 at a significance level of 0.01. Because of this high correlation between Quality and Noise scores, only the Quality scores have been used in the subsequent analysis.

Table 6.1: *Correlation between Quality and selected evaluation measures*

	Corr. experiment 1	Corr. experiment 2
CPR	0.610	0.789
MB	-0.509	-0.375
SDM	0.453	0.498
Mdis	-0.432	0.001
ST	0.27	0.296
(P)SNRI	0.258	0.152
DPC	-0.254	0.068
TDM	0.239	0.275
MSE	-0.161	-0.09
FFNSA	-0.15	0.086
MD	-0.133	-0.095
PSNR	0.103	-0.082
SNR	0.091	0.048
WSNR2	-0.082	-0.078
WPSNR2	-0.062	-0.082
WSNR1	-0.044	-0.046
WSNR3	-0.044	-0.052
WPSNR3	-0.027	-0.053
WPSNR1	-0.022	-0.046

6.2 Evaluation measures that correlate with quality

The correlation between the selected objective and hybrid evaluation measures has been analysed by calculating the Pearson (2-tailed) correlation between the averaged Quality score and the evaluation measures. The results for experiment 1 and 2 are given in Table 6.1. Only the objective measures are selected that have a correlation coefficient in excess of 0.40. Thus, the *CPR*, *MB* and *SDM* evaluation measures were selected. From the table, it is clear that the *Mdis* evaluation measure almost does not correlate with Quality in experiment 2. This evaluation measure is, therefore, discarded for the rest of the analysis.

6.3 Correlation among the selected evaluation measures

The correlation among the evaluation measures that were selected in the previous paragraph has been calculated with a Pearson's (2-tailed) correlation analysis. Results were for experiment 1 and 2, that the *MB* evaluation measure correlates highly with the *SDM* evaluation measure, with correlation coefficient -0.917 at significance level < 0.01. For the other evaluation measures a correlation could not be determined at a significant level. Therefore, the *SDM*

evaluation measure has been removed from the analysis.

6.4 Regression model of subjective quality

With the two remaing evaluation measures, that both correlated with Quality and are not highly correlated with each other, two regression models have been calculated using SPSS. The averaged Quality score of experiment 1 over the three sequences was used as dependent variable and the (normalised)[1] *CPR* and *MB* evaluation measures were used as independent variables. The model, with standardised coefficients, for experiment 1 was (Equation 6.1)

$$Subjective\ quality = 0.607 * CPR_{norm.} - 0.505 * MB_{norm.} \qquad (6.1)$$

The adjusted R^2 is 0.503 and the F-test shows a significance level of 0.052 (F = 5.052, df = 2). The Durbin-Watson test for serial correlation of the residuals is 2.528, meaning that the residuals are almost uncorrelated.

For experiment 2, the same regression analysis has been performed with the averaged Quality score over the three sequences as dependent variable and the (normalised) *CPR* and *MB* evaluation measures as independent variables. The model, with standardised coefficients, for experiment 2 was (Equation 6.2)

$$Subjective\ quality = 0.870 * CPR_{norm.} - 0.511 * MB_{norm.} \qquad (6.2)$$

The adjusted R^2 is 0.837 and the F-test shows a significance level of 0.002 (F = 21.547, df = 2). The Durbin-Watson test for serial correlation of the residuals is 2.692, which shows that the residuals are almost uncorrelated.

For experiment 1 and 2, the data files have been merged casewise. On these data, a combined regression analysis has been performed with the averaged Quality score over the three sequences as dependent variable and the (normalised) *CPR* and *MB* evaluation measures as independent variables. The model, with standardised coefficients, for experiment 1 and 2 together was (Equation 6.3)

$$Subjective\ quality = 0.664 * CPR_{norm.} - 0.497 * MB_{norm.} \qquad (6.3)$$

The adjusted R^2 is 0.593 and the F-test shows a significance level of < 0.001 (F = 13.395, df = 2). The Durbin-Watson test for serial correlation of the residuals is 2.305, so the residuals are almost uncorrelated.

6.5 Conclusions

The scores for Quality and Noise have been averaged over the three sequences. It appeared from a correlation analysis with SPSS that the averaged Quality

[1]Normalisation of the evaluation score means calculating $\frac{ev.score - av.score}{std.dev.\ per\ experiment}$.

and Noise scores are highly correlated. The correlation coefficients of all objective and hybrid evaluation measures with Quality have been calculated. The results show that correlation in general of the objective and hybrid evaluation measures with quality as measured in the two subjective experiments is very low. The best correlating independent measures are *CPR* and *MB*. These evaluation measures have been used for the two regression models of the results of experiment separately, and for the regression model of the combined results of the two experiments. The regression model of the first experiment gives a very low explanatory power of 50% (R^2 square value of 0.503) of the variance of the subjective quality. The regression model of the second experiment gives a satisfactory explanatory power of 84% (R^2 square value of 0.837). The regression model for the combined results of experiment 1 and 2 gives a low explanatory power of 60% (R^2 square value of 0.593). The coefficients of the three regression models are close together, which means that the model is consistent for the individual experiments and the combined experimental results. The explanatory power of the model for the combined experimental results is with 60% very low.

Chapter 7

Conclusions

In this report, three assignments needed to be completed. First, noise reduction algorithms that are suitable for consumer applications, given their complexity, architecture and performance had to be found. Second, the noise reduction algorithms had to be evaluated with the help of a subjective quality experiment and a correlation of the results with objective quality measures had to be explored. Finally, recommendations for a video noise reduction quality measure to improve the objective evaluation of video noise reduction algorithms had to be given.
To comply to these assignments, different characteristics of noise reduction algorithms have been described to find suitable noise algorithms. A description of filter support, support size, filter structure types, noise filter features, neighbourhood selection and adaptivity, statistics used in filtering and complexity and hardware constraints has been given. Furthermore, different spatial and temporal filters have been described. In addition, one combined spatial-temporal filter has been described too. Subsequently, seven (combinations of) noise reduction algorithms have been selected for evaluation. These filters are: Falcon_mc_hp, Siemens, SWAN_DNR, Median_Fuzzy, SWAN, DNR and DNR_mc.
These algorithms have been implemented and used in three different evaluation methods. The results of these methods will be given in the following paragraphs. The results have been analysed thoroughly, among others the correlation among objective and hybrid evaluation measures has been considered. Also, the correlation between subjective and objective and hybrid evaluation measures has been investigated. Thus, the second assignment is fulfilled.
In the end of this chapter, results to obtain a better noise evaluation measure have been given.

7.1 Evaluation methods

The evaluation of the selected noise reduction algorithms has been done by objective, hybrid and subjective evaluation measures. Several objective, hybrid and subjective evaluation measures have been explained in detail. The objective measures are: *SNR*, *PSNR*, *SNRI/PSNRI*, *MSE*, *MD*, *MB*, *ST*, *CPR*, *FFNSA* and *DPC*. The hybrid evaluation measures are: Fujio *SNR* and *PSNR*, CCIR

4211 *SNR* and *PSNR*, CCIR 5672 *SNR* and *PSNR*, *SDM*, *TDM* and *Mdis*. These measures have been used for the objective (and hybrid) evaluation of seven noise reduction algorithms.

7.2 Objective and hybrid evaluation results

The noise algorithms are ranked on the overall performance for 26 and 32 dB *PSNR* sequences. Also an overview has been given of all objective and hybrid evaluation measures (Table 5.1 and Table 5.2). The ranking of the algorithms is for the 26 dB *PSNR* sequences, in order of goodness: SWAN_DNR, Siemens, DNR_mc, SWAN, Median_Fuzzy, DNR and Falcon_mc_hp. For 32 dB *PSNR* sequences, in order of goodness, the ranking is: SWAN_DNR, SWAN, DNR_mc, Median_Fuzzy, DNR, Siemens and Falcon_mc_hp. From the results, it became clear that there was a strong consistency for *SNR*, *PSNR*, *SNRI*, *MD*, *MSE* and *MB* evaluation measures on 26 dB *PSNR* sequences. For 32 dB *PSNR* sequences, the same group of evaluation measures was consistent among each other, only small differences arose for the *MSE* evaluation measure, while the ranking of the noise reduction algorithms of the *MB* evaluation measure deviated heavily.
Another group of evaluation measures that shows a highly consistent behaviour is the group of the weighted *SNR* and *PSNR* evaluation measures. There were only small deviations within this group.

7.3 Subjective evaluation results

The conducted subjective test containing two experiments gave results on quality and noise performance of the noise reduction algorithms and on important parts that observers use to evaluate quality and noise performance.

Overall in experiment 1, the Siemens algorithm was the best, for both quality as noise performance. Next came SWAN_DNR together with Falcon_mc_hp. SWAN could not be distinguished on noise performance from Falcon_mc_hp. On quality performance, SWAN was worse than SWAN_DNR and Falcon_mc_hp. The worst performing algorithm for both quality and noise performance was the Median_Fuzzy algorithm.

Subsequently, the effect of noise strength on quality and noise performance has been described. This is done by examining the quality and noise performance for 26 dB and for 32 dB *PSNR* sequences separately.
For 26 dB *PSNR* sequences, the Siemens algorithm performed the best, followed by the SWAN_DNR algorithm, then came Falcon_mc_hp and SWAN that were comparably good. The worst algorithm was Median_Fuzzy.
For 32 dB *PSNR* sequences, the Siemens noise reduction algorithm was the best perceived algorithm, next came the Falcon_mc_hp and the SWAN_DNR algorithms and, finally, the SWAN_DNR and Median_Fuzzy algorithms. The SWAN_DNR algorithm had a combined second and third rank, together with,

respectively, Falcon_mc_hp and Median_Fuzzy. Results for quality and noise performance for the two different noise strengths were exactly the same.

In experiment 2, the SWAN_DNR, DNR_mc and Falcon_mc_hp algorithms scored the best. These algorithms were undistinguishable on basis of quality performance. Next came the SWAN algorithm, followed by the DNR noise reduction algorithm that performed the worst on quality. The noise performance was the same, only Falcon_mc_hp and SWAN algorithms could not be distinguished from each other. For both noise and quality performance of noise reduction on 26 dB *PSNR* sequences, SWAN_DNR scored the best (on noise SWAN_DNR was comparably good as DNR_mc). DNR performed the worst for both scores. DNR_mc and Falcon_mc_hp scored comparably for both quality and noise performance and were better performing than DNR. For both quality and noise performance for 32 dB *PSNR* sequences, DNR_mc, SWAN_DNR, Falcon_mc_hp and SWAN scored better than the DNR noise reduction algorithm. DNR scored the worst for quality and noise.

The observers were also asked to indicate important areas for judgement of quality and noise. Results showed that for the sequences *Renata* and *Car & Gate* no explicit preference can be given for detailed or undetailed parts. For the *Teeny* sequence a clear preference is present for undetailed areas of the sequence. These results are applicable for quality as well as noise judgement.

7.4 Regression model of subjective quality

The scores for quality and noise performance obtained from the two subjective experiments have been averaged over the three sequences. It appeared from a correlation analysis with SPSS that the averaged quality and noise scores are highly correlated. The correlation coefficients of all objective and hybrid evaluation measures with the quality performance have been calculated. The results show that correlation in general of the objective and hybrid evaluation measures with quality as measured in the two subjective experiments is very low. The best correlating independent measures are *CPR* and *MB*. These evaluation measures have been used for the two regression models of the results of experiment separately, and for the regression model of the combined results of the two experiments. The regression model of the first experiment gives a very low explanatory power of 50% (R^2 square value of 0.503) of the variance of the subjective quality. The regression model of the second experiment gives a satisfactory explanatory power of 84% (R^2 square value of 0.837). The regression model for the combined results of experiment 1 and 2 gives a low explanatory power of 60% (R^2 square value of 0.593). The coefficents of the three regression models are close together, which means that the model is consistent for the individual experiments and the combined experimental results. The explanatory power of the model for the combined experimental results is with 60% very low.

References

[1] G. de Haan. *Video processing for multimedia systems.* University Press Facilities Eindhoven, 2000.

[2] J. Allnatt. *Transmitted-picture assessment.* John Wiley & Sons Ltd., 1983.

[3] K. Jostschulte and A. Amer. A new cascaded spatio-temporal noise reduction scheme for interlaced video. In *Proceedings of International Conference on Image Processing*, pages 493–497. IEEE, Oct. 1998.

[4] K. Jostschulte et al. Perception adaptive temporal tv-noise reduction using contour preserving pre-filter techniques. *IEEE Transactions on Consumer Electronics*, 44(3):1091–1096, Aug. 1998.

[5] A. Amer and H. Schröder. A new video noise reduction algorithm using spatial subbands. In *Proceedings of the Third IEEE International Conference on Electronics, Circuits, and Systems*, pages 45–48. IEEE, Oct. 1996.

[6] K. Jostschulte et al. A new video noise reduction algorithm using spatial subbands. In *Proceedings of International Conference on Consumer Electronics*, pages 438–439. IEEE, June 1998.

[7] O.A. Ojo and T.G. Kwaaitaal-Spassova. An algorithm for integrated noise reduction and sharpness enhancement. *IEEE Transactions on Consumer Electronics*, 46(3):474–480, Aug. 2000.

[8] M. Mancuso et al. Fuzzy edge-oriented motion-adaptive noise reduction and scanning rate conversion. In *Asia-Pacific Conference on Circuits and Systems*, pages 652–656. IEEE, Dec. 1994.

[9] M. Mancuso et al. Fuzzy logic based image processing in iqtv environment. *IEEE Transactions on Consumer Electronics*, 41(3):917–925, Aug. 1995.

[10] T. Grafe and G. Scheffler. Interfield noise and cross color reduction ic for flicker free tv receivers. *IEEE Transactions on Consumer Electronics*, 34(3):402–408, Aug. 1988.

[11] G. de Haan et al. Automatic 2-d and 3-d noise filtering for high-quality television. In *Proceedings of the 7th International Workshop in HDTV*, Oct. 1994.

[12] CCIR. Avis 421-1, annex iii. In *Docements de la XIe assemblé plénière, Oslo, 1966, Volume V*, pages 81–82, Geneva, 1967.

[13] CCIR Recommendation. 567-2, part c, annex ii, par. 3. In *Recommendations and reports of the CCIR XVIth plenary assembly, Dubrovnik, 1986, Volume XII*, pages 24–25, Geneva, 1986.

[14] T. Fujio. *A Universal noise weighting function and its application to high-definition television system design.* NHK Laboratories Note, No. 240, Sept. 1979.

[15] V. Kayargadde and J.-B. Martens. An objective measure for perceived noise. *Signal Processing*, 49:187–206, 1996.

[16] S. Voran. The development of objective video quality measures that emulate human perception. In *Globecom '91*, pages 1776–1781. IEEE, 1991.

[17] S.D. Voran and S. Wolf. The development and evaluation of an objective video quality assessment system that emulates human viewing panels. In *International Broadcasting Convention*, pages 504–508. IEEE, 1992.

[18] W.Y. WU et al. Performance evaluation of some noise-reduction methods. *Computer Vision, Graphics, and Image Processing*, 54(2):134–146, March 1992.

[19] G.A. Mastin. Adaptive filters for digital image noise smoothing: an evaluation. *Computer Vision, Graphics, and Image Processing*, 31:103–121, 1985.

[20] ITU-R Recommendation BT.500-7. Methodology for the subjective assessment of the quality of television pictures, 1996.

[21] Peter G. Engeldrum. *Psychometric Scaling: A Toolkit for Imaging Systems Development.* Imcotek Press, 2000.

Appendix A

Objective measurement results

The results of the *SNR* measure are given in Figure A.1 and Figure A.2. It is seen, that results for *SNR* and *PSNR* are almost the same. The results of the SWAN and DNR_mc algorithms in the 32 dB *PSNR* sequences are slightly different.

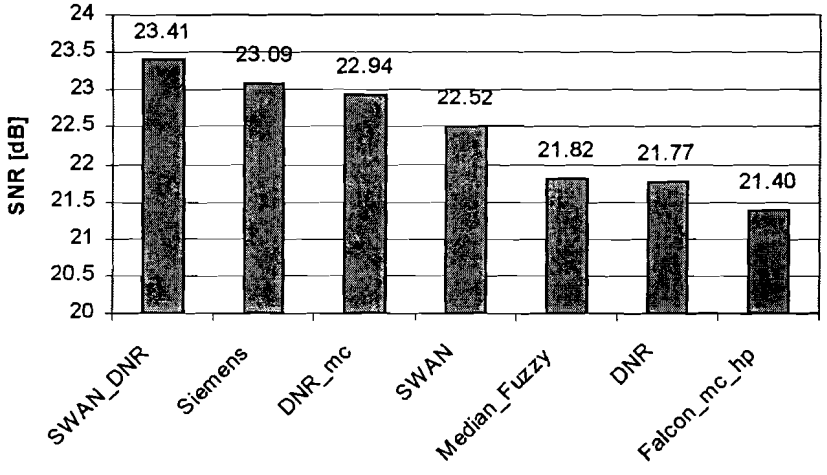

Figure A.1: *SNR measure evaluation for seven algorithms on 26 dB PSNR sequences*

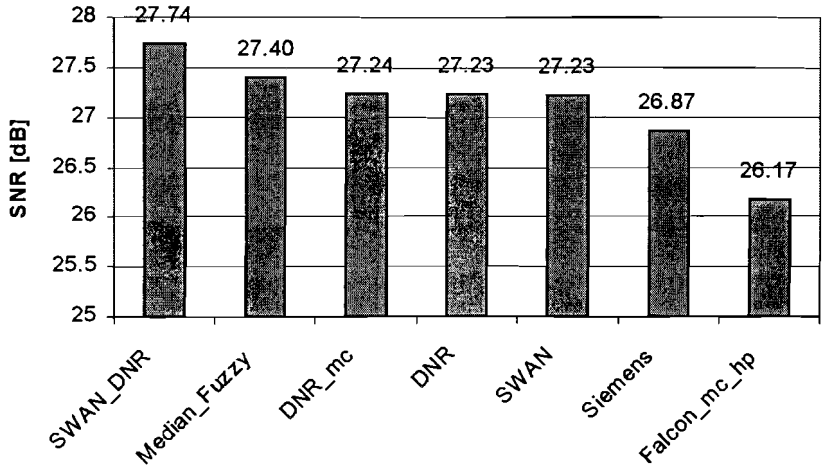

Figure A.2: *SNR measure evaluation for seven algorithms on 32 dB PSNR sequences*

In Figure A.3 and Figure A.4, the results are given of the *PSNR* measure for, 26 dB and 32 *PSNR* noise degradations, respectively. It can be seen that the SWAN_DNR algorithm performs the best, while the Falcon_mc_hp algorithm performs the worst. The *PSNRI/SNRI* evaluation measure gives the same results, as can be seen in Figures A.5 and Figure A.6.

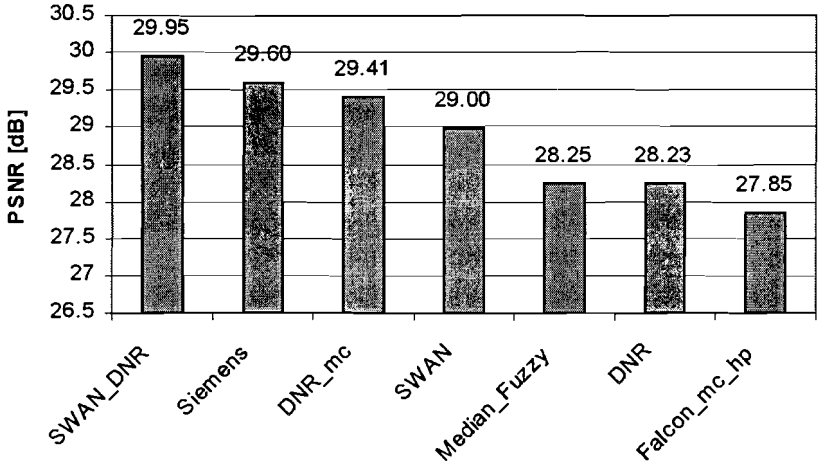

Figure A.3: *PSNR measure evaluation for seven algorithms on 26 dB PSNR sequences*

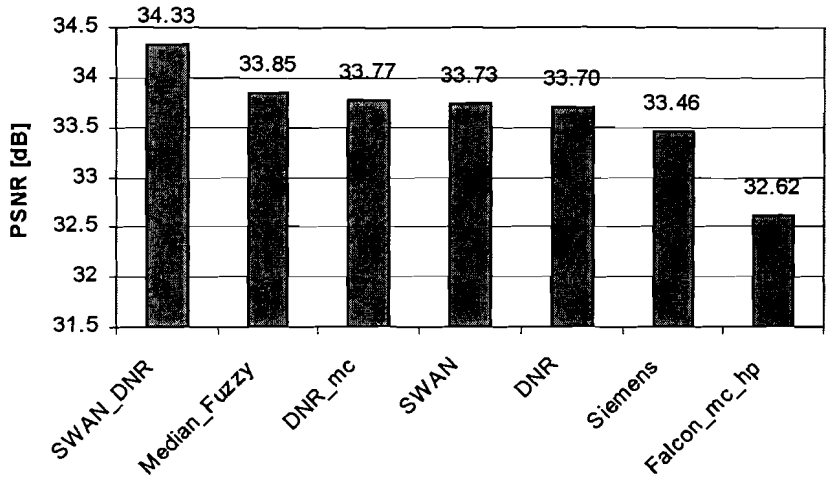

Figure A.4: *PSNR measure evaluation for seven algorithms on 32 dB PSNR sequences*

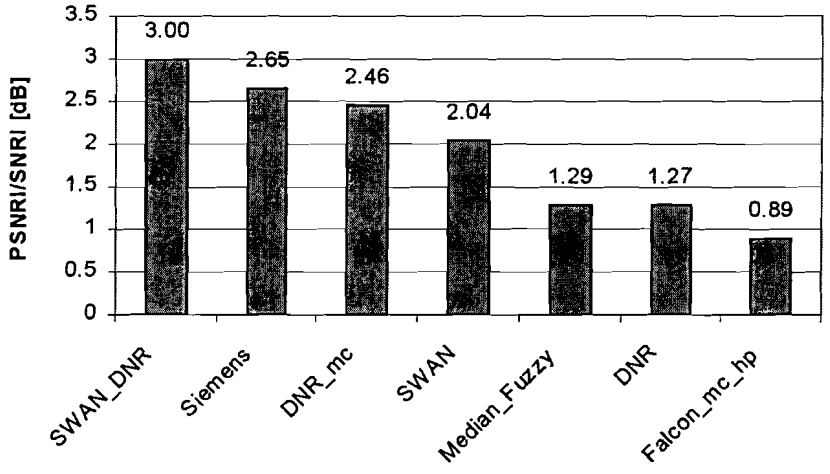

Figure A.5: *PSNRI measure evaluation for seven algorithms on 26 dB PSNR sequences*

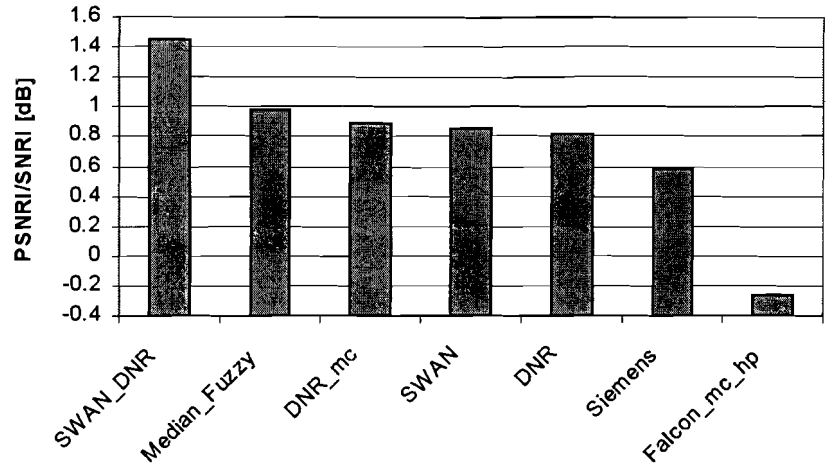

Figure A.6: *PSNRI measure evaluation for seven algorithms on 32 dB PSNR sequences*

In Figure A.7 and Figure A.8, the evaluation results for the MD/MAE measures is given. The results are exactly the same as the $PSNR$ evaluation measure.

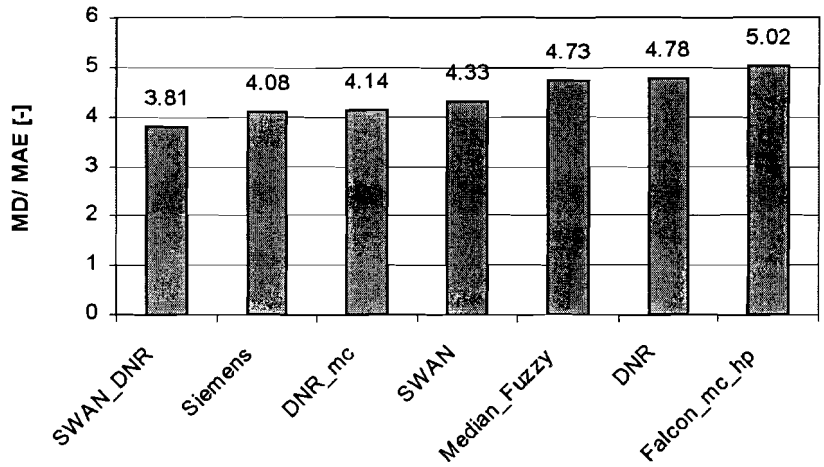

Figure A.7: *MD/MAE measure evaluation for seven algorithms on 26 dB PSNR sequences*

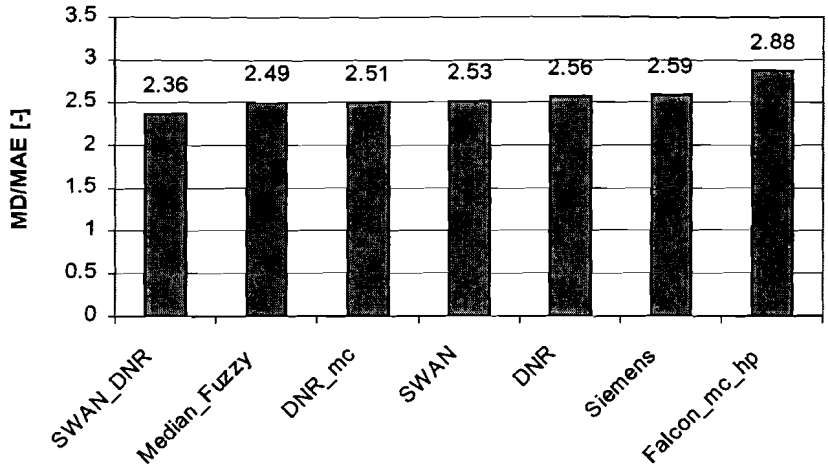

Figure A.8: *MD/MAE measure evaluation for seven algorithms on 32 dB PSNR sequences*

In Figure A.9 and Figure A.10, the *MSE* evaluation results are summarised. Results are the same as the *SNR* evaluation measure, only the results for the DNR, DNR_mc and SWAN algorithms differ in ranking.

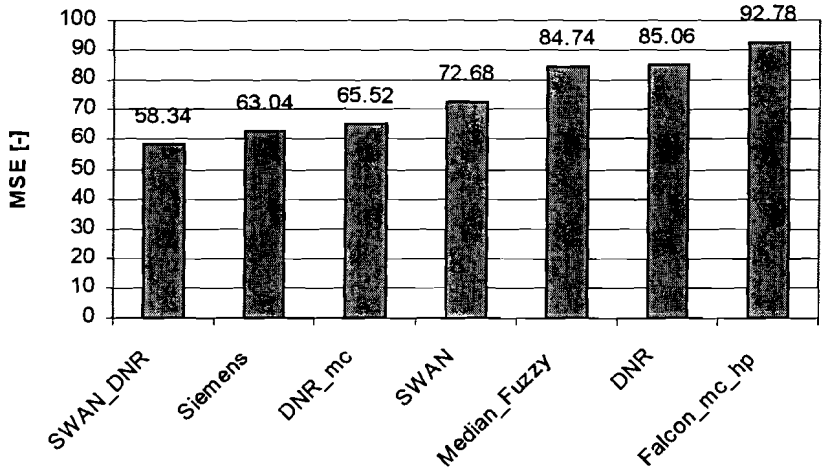

Figure A.9: *MSE measure evaluation for seven algorithms on 26 dB PSNR sequences*

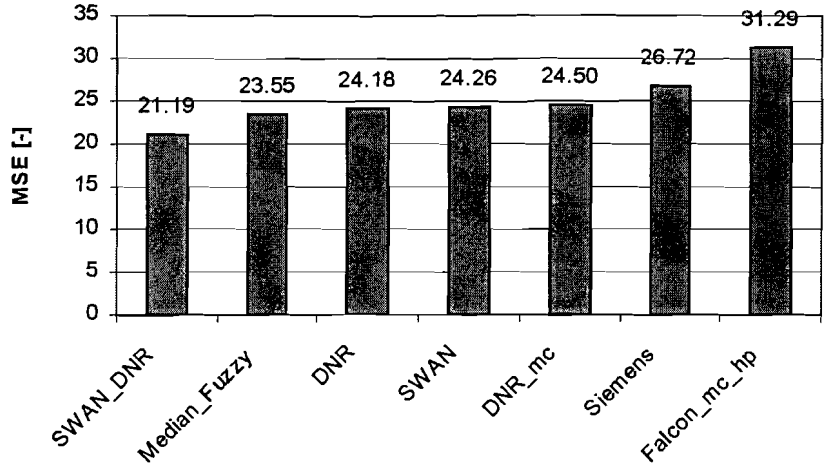

Figure A.10: *MSE measure evaluation for seven algorithms on 32 dB PSNR sequences*

The *MB* evaluation results are given in Figure A.11 and Figure A.12. The Siemens algorithm has the lowest Mean Busyness value for both noise degradations, while the DNR and Falcon_mc_hp algorithms have the highest *MB* values.

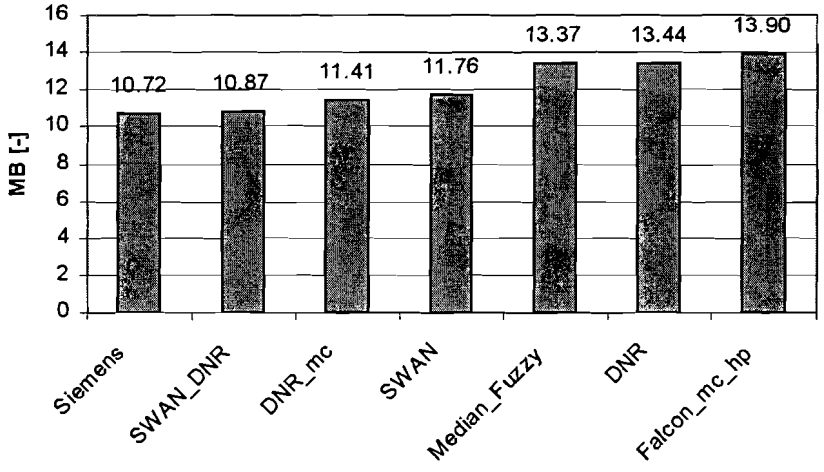

Figure A.11: *MB measure evaluation for seven algorithms on 26 dB PSNR sequences*

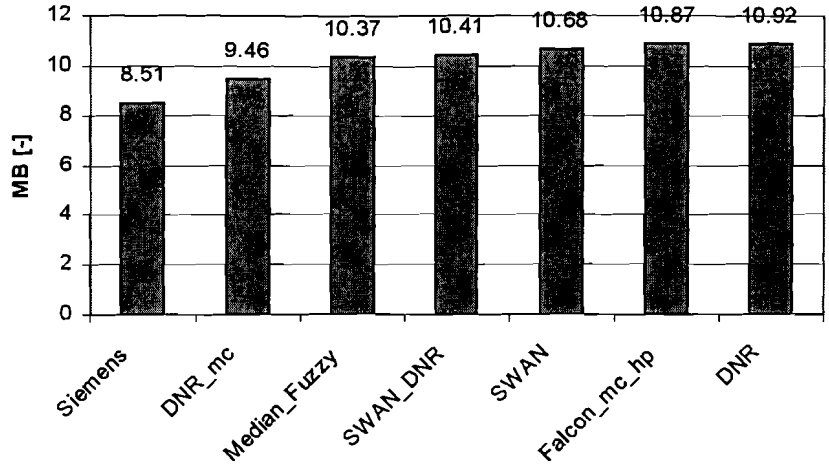

Figure A.12: *MB measure evaluation for seven algorithms on 32 dB PSNR sequences*

The stability (ST) results in Figure A.13 and Figure A.14 show that algorithm Siemens scores good in both noise degradations, number two and one respectively, while the Falcon_mc_hp algorithm has low ST scores, last place and sixth place, respectively.

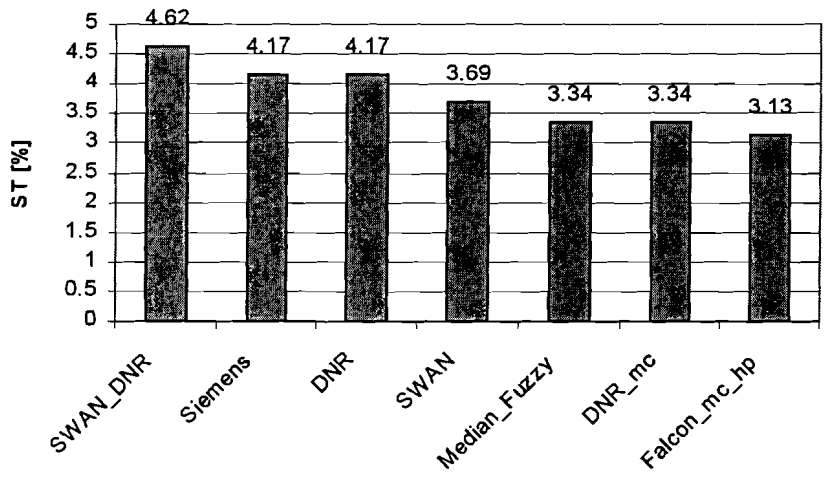

Figure A.13: *ST measure evaluation for seven algorithms on 26 dB PSNR sequences*

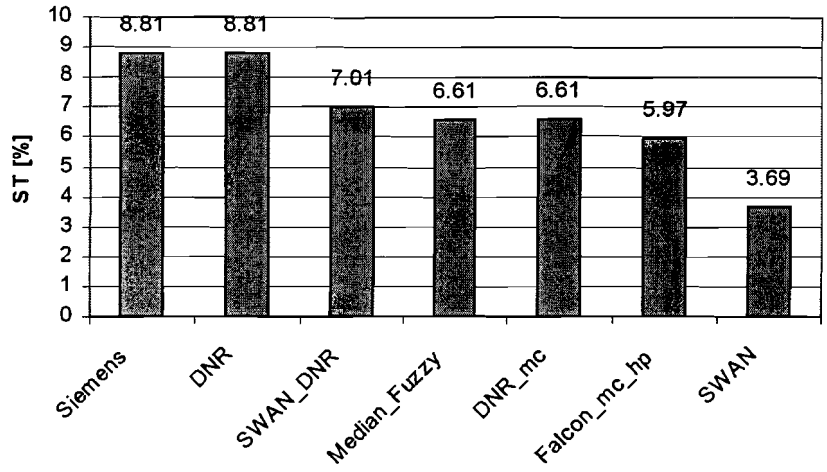

Figure A.14: *ST measure evaluation for seven algorithms on 32 dB PSNR sequences*

Correct processing ratio measures (*CPR*) are given in Figure A.15 and Figure A.16. Falcon_mc_hp, DNR_mc, SWAN_DNR and Siemens algorithms score all high for both noise degradations. The Median_Fuzzy and DNR algorithms score low for both noise degradations. SWAN is somewhere in between these two groups.

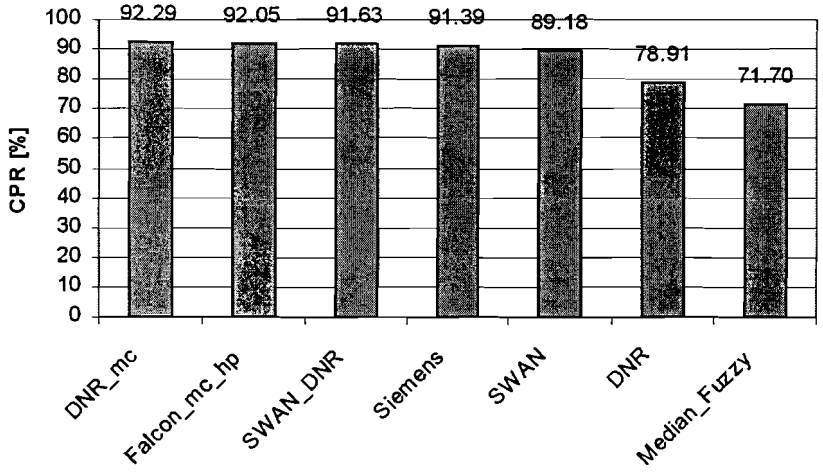

Figure A.15: *CPR measure evaluation for seven algorithms on 26 dB PSNR sequences*

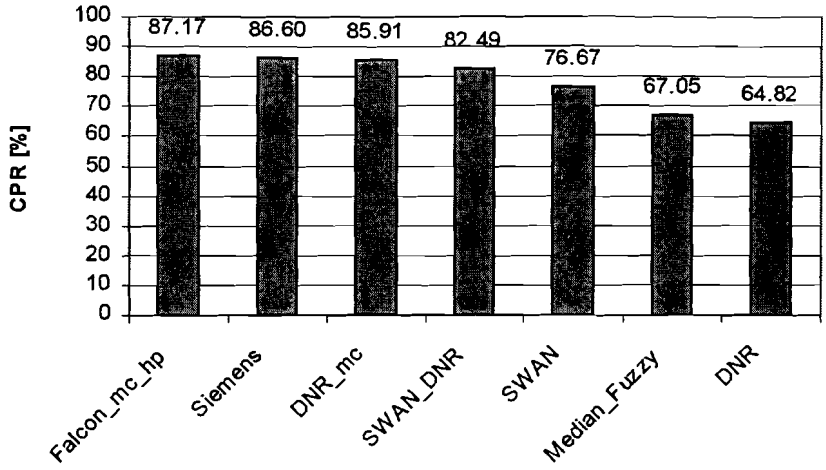

Figure A.16: *CPR measure evaluation for seven algorithms on 32 dB PSNR sequences*

The Flat Field Noise Smoothing Ability is described in Figure A.17 and Figure A.18. It can be seen that for 26 dB (PSNR) sequences, the DNR_mc algorithm performs the best and the SWAN algorithm the worst. For 32 dB (PSNR) sequences, the SWAN_DNR algorithm was the best and the Siemens algorithm the worst.

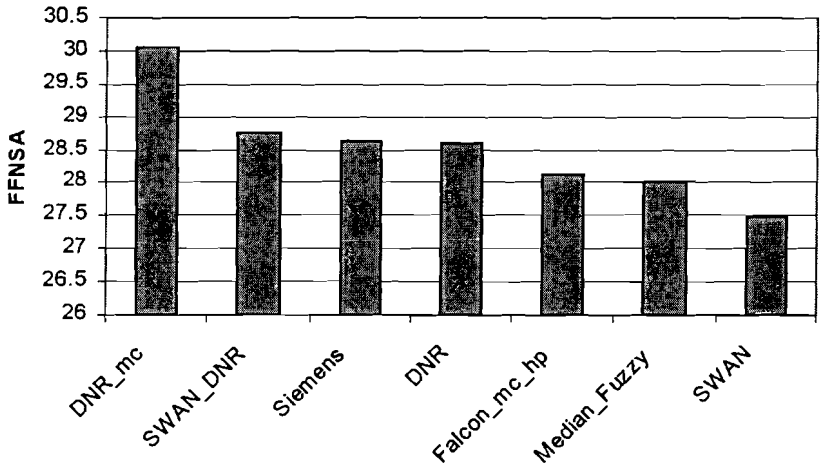

Figure A.17: *FFNSA measure evaluation for seven algorithms on 26 dB PSNR sequences*

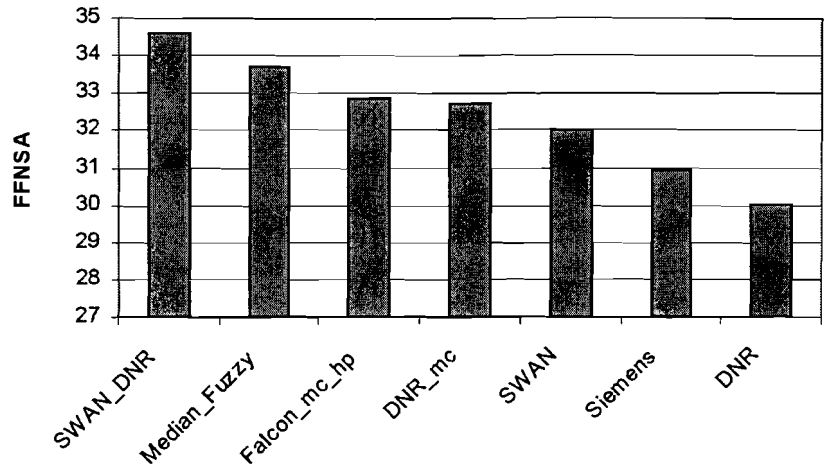

Figure A.18: *FFNSA measure evaluation for seven algorithms on 32 dB PSNR sequences*

The Detail-Preservation Capability is described in Figure A.19 and Figure A.20 for the two noise strengths. For the 26 dB (PSNR) sequences, the SWAN_DNR algorithm performs the best and the Falcon_mc_hp the worst. For 32 dB (PSNR) sequences, the Siemens algorithm was the best and the Falcon_mc_hp the worst.

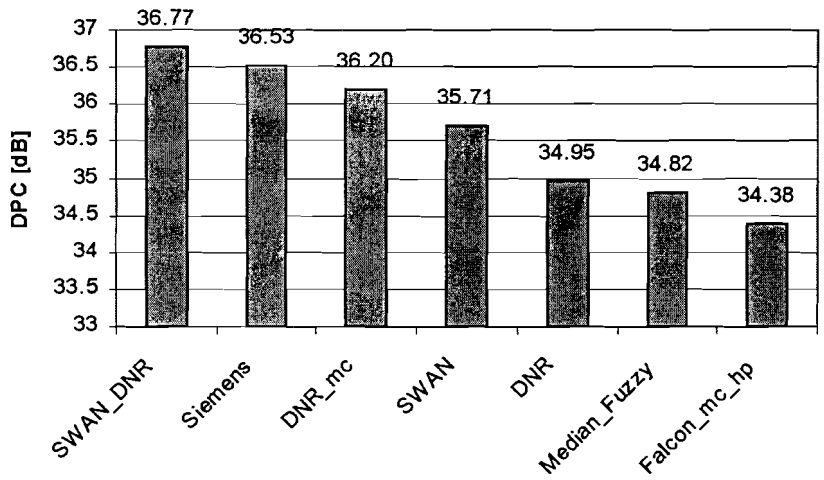

Figure A.19: *DPC measure evaluation for seven algorithms on 26 dB PSNR sequences*

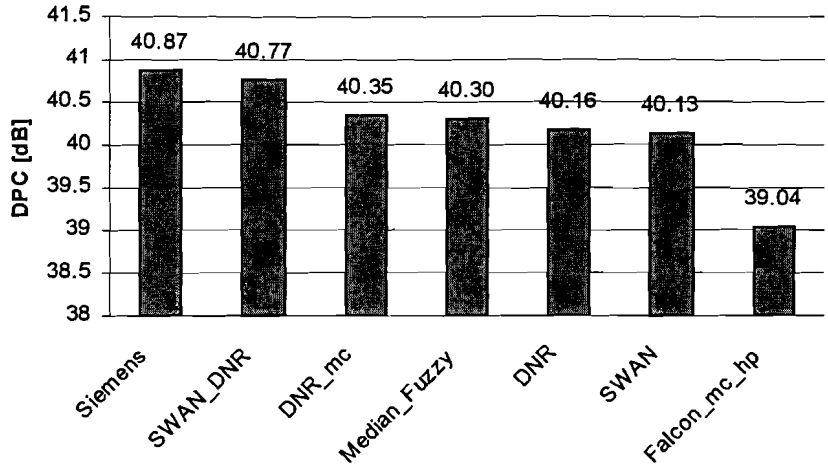

Figure A.20: *DPC measure evaluation for seven algorithms on 32 dB PSNR sequences*

Appendix B

Hybrid measurement results

In Figure B.1 and B.2, the results of the Fujio *PSNR* evaluation measure are given. It can be seen that the SWAN_DNR and the SWAN algorithm perform the best and Falcon_mc_hp performs the least good.

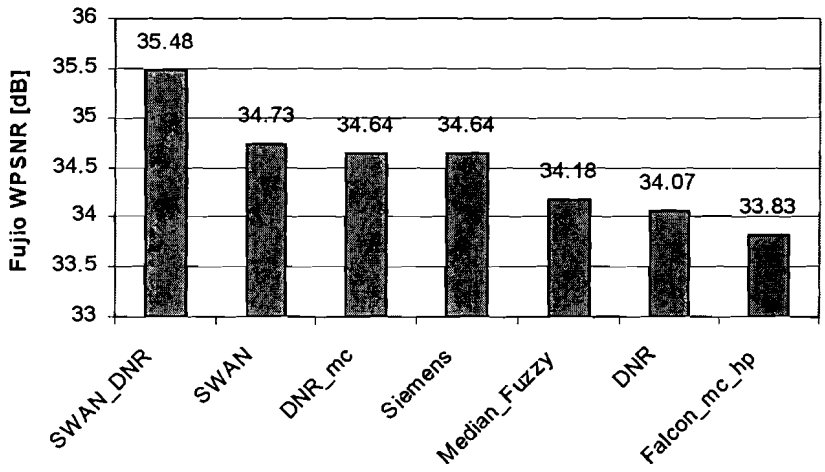

Figure B.1: *Fujio PSNR measure evaluation for seven algorithms on 26 dB PSNR sequences*

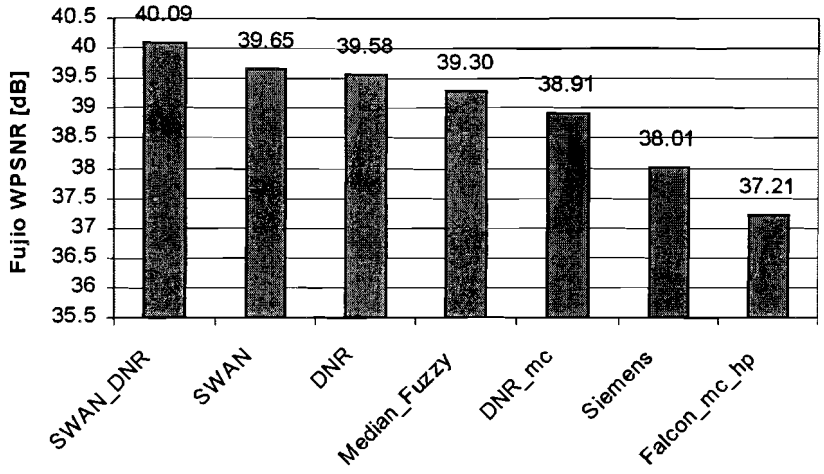

Figure B.2: *Fujio PSNR measure evaluation for seven algorithms on 32 dB PSNR sequences*

In Figure B.3 and B.4, the Fujio *SNR* evaluation measure gives the same results, as the Fujio *PSNR* evaluation measure.

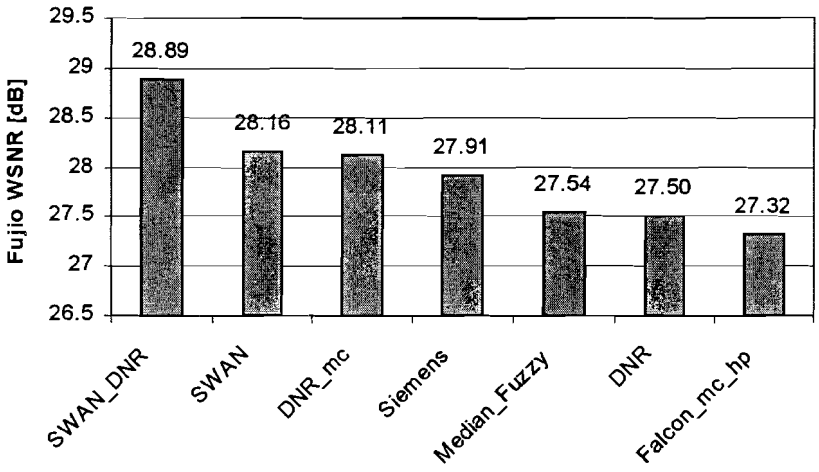

Figure B.3: *Fujio SNR measure evaluation for seven algorithms on 26 dB PSNR sequences*

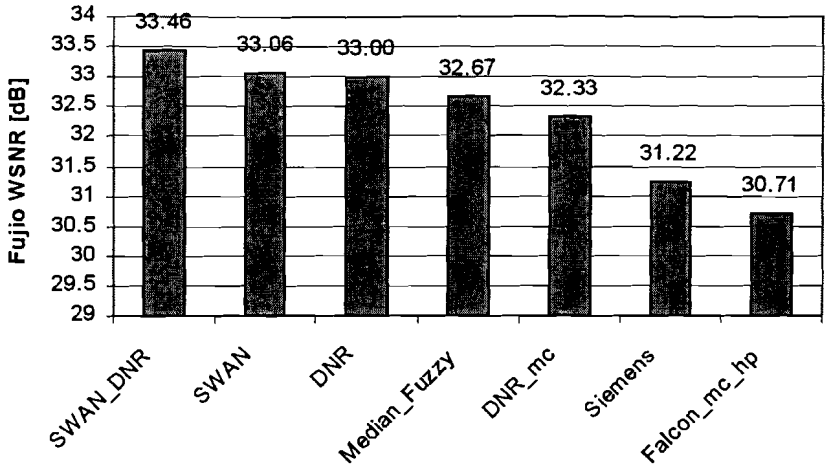

Figure B.4: *Fujio SNR measure evaluation for seven algorithms on 32 dB PSNR sequences*

In Figure B.5 and B.6, the results of the CCIR 4211 *PSNR* evaluation measure are given. The results are the same as the Fujio *PSNR* and *SNR* evaluation results.

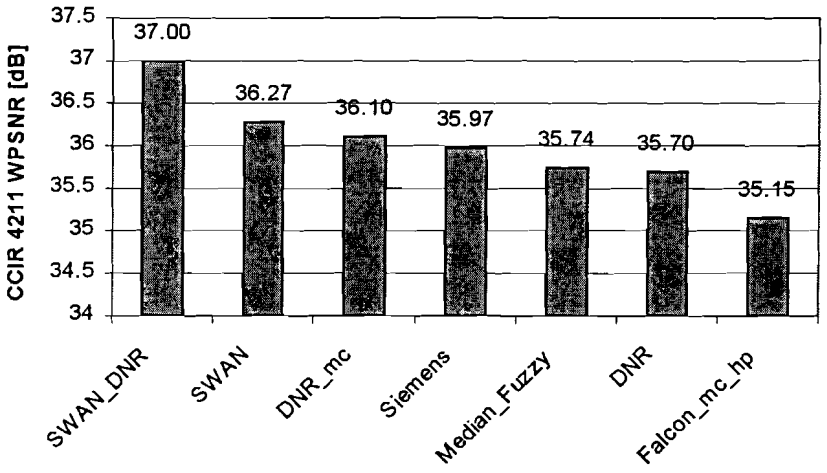

Figure B.5: *CCIR 4211 PSNR measure evaluation for seven algorithms on 26 dB PSNR sequences*

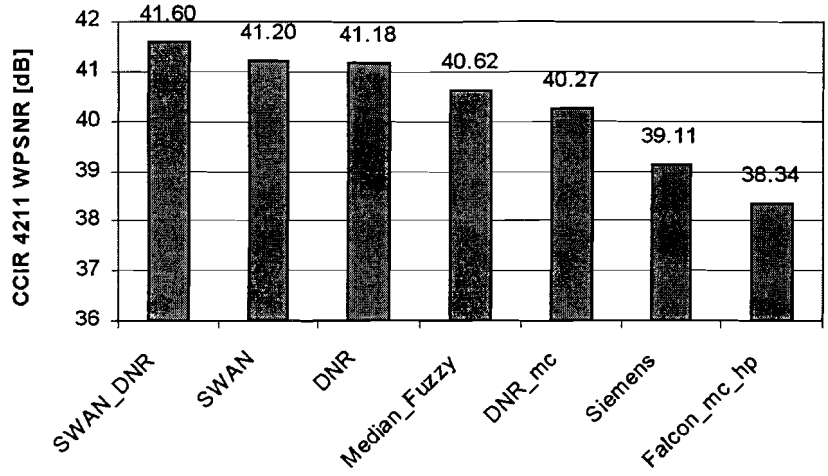

Figure B.6: *CCIR 4211 PSNR measure evaluation for seven algorithms on 32 dB PSNR sequences*

In Figure B.7 and B.8, the results of the CCIR 4211 *SNR* evaluation measure are given. The results are similar to the results of the Fujio *PSNR*, Fujio *SNR* and CCIR 4211 *PSNR* results, apart from the results for the 26 dB *PSNR* sequences where different rankings occur for the Median_Fuzzy and DNR algorithms. The evaluation measures values for these algorithms are very close together.

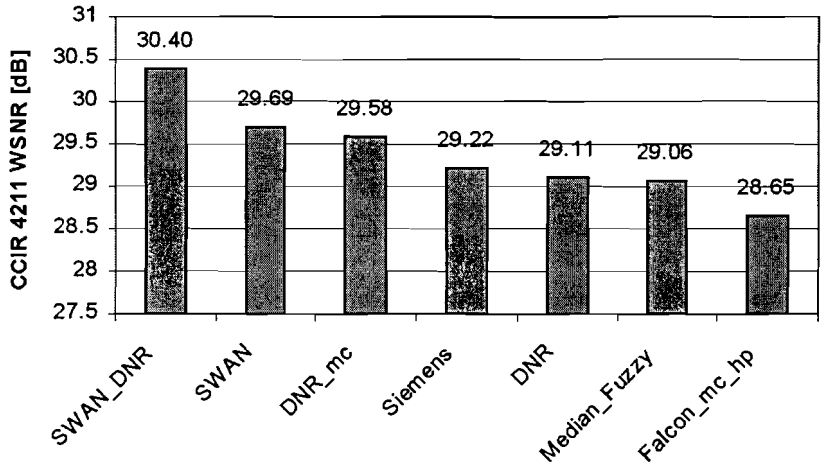

Figure B.7: *CCIR 4211 SNR measure evaluation for seven algorithms on 26 dB PSNR sequences*

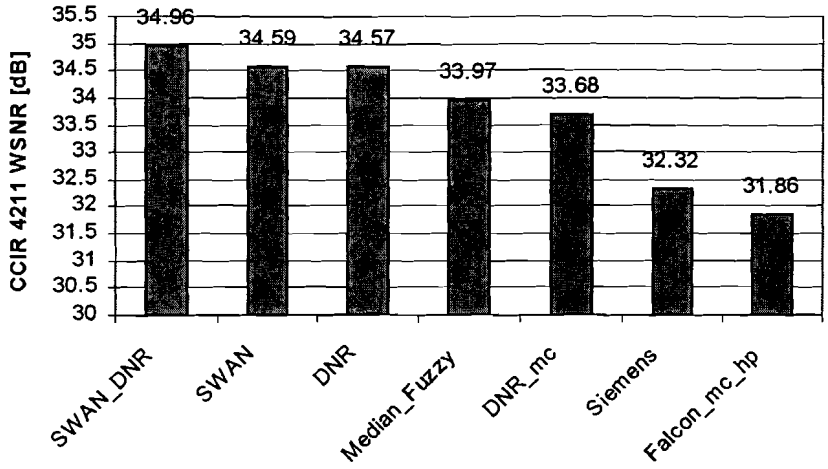

Figure B.8: *CCIR 4211 SNR measure evaluation for seven algorithms on 32 dB PSNR sequences*

In Figure B.9, Figure B.10, Figure B.11 and Figure B.12, the CCIR 5672 *PSNR* and *SNR* measures evaluation results are given. Results are very similar to the previous given weighted *PSNR* and *SNR* measures.

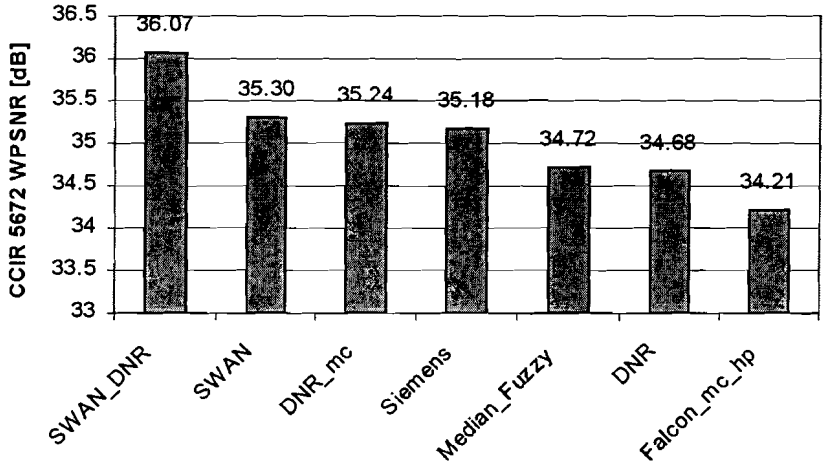

Figure B.9: *CCIR 5672 PSNR measure evaluation for seven algorithms on 26 dB PSNR sequences*

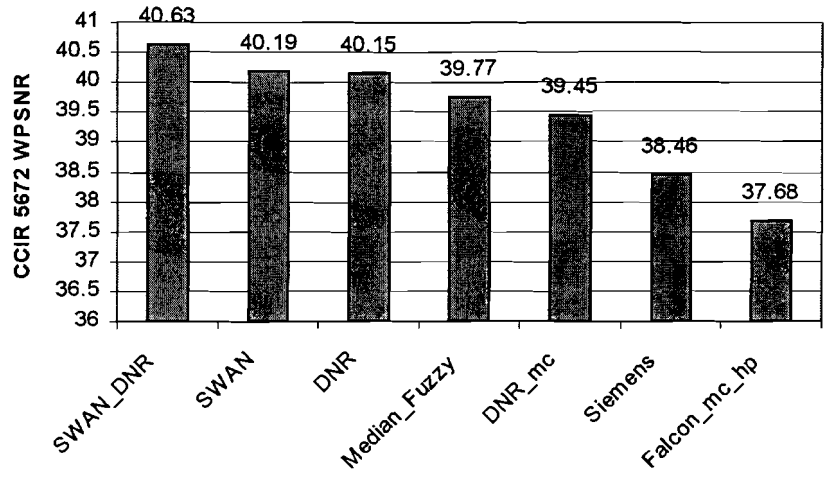

Figure B.10: *CCIR 5672 PSNR measure evaluation for seven algorithms on 32 dB PSNR sequences*

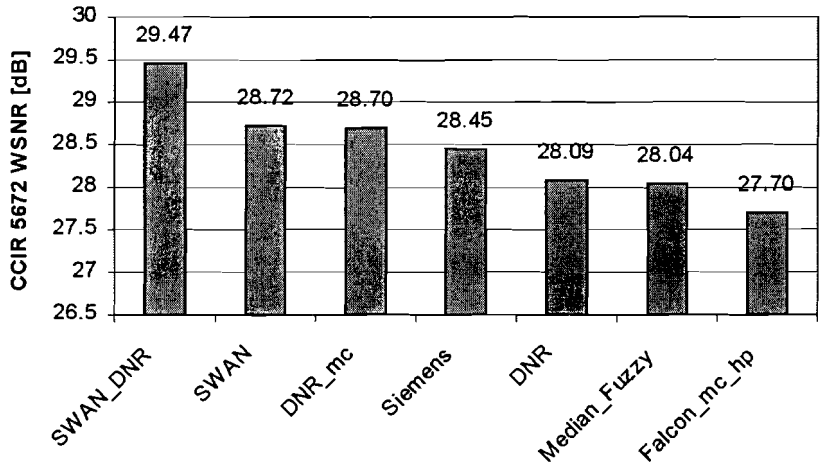

Figure B.11: *CCIR 5672 SNR measure evaluation for seven algorithms on 26 dB PSNR sequences*

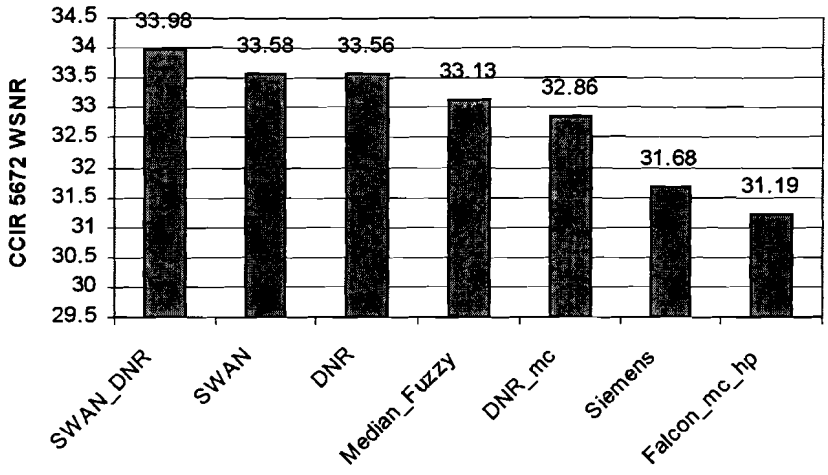

Figure B.12: *CCIR 5672 SNR measure evaluation for seven algorithms on 32 dB PSNR sequences*

In Figure B.13 and Figure B.14, the *SDM* evaluation measure results are given. For 26 dB *PSNR* sequences, SWAN_DNR performs the best, while Falcon_mc_hp performs the worst for 26 dB *PSNR* sequences. For the 32 dB *PSNR* sequences, the Siemens algorithm performs the best and the Falcon_mc_hp algorithm the worst.

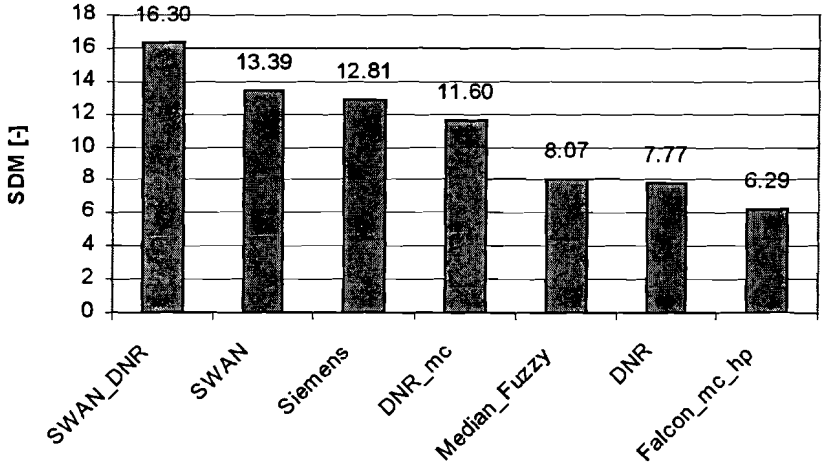

Figure B.13: *SDM measure evaluation for seven algorithms on 26 dB PSNR sequences*

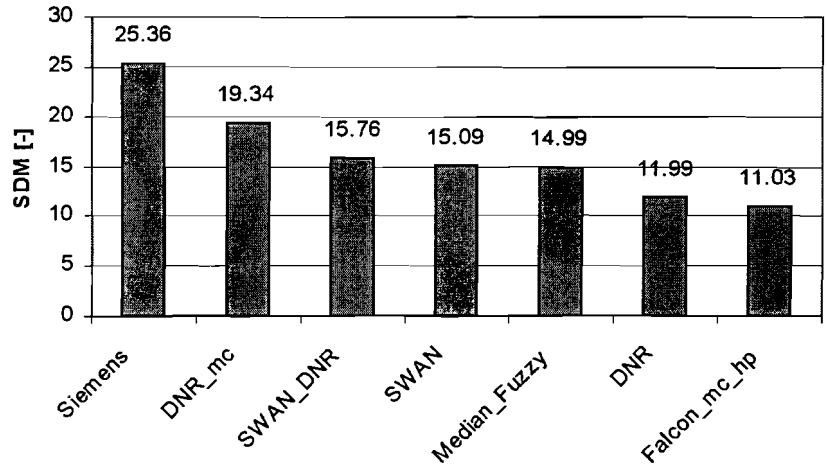

Figure B.14: *SDM measure evaluation for seven algorithms on 32 dB PSNR sequences*

In Figure B.15 and Figure B.16, the *TDM* measure results are given. The Siemens algorithm performs the best for both the 26 dB as the 32 dB *PSNR* sequences. The DNR algorithm performs the worst for both noise levels.

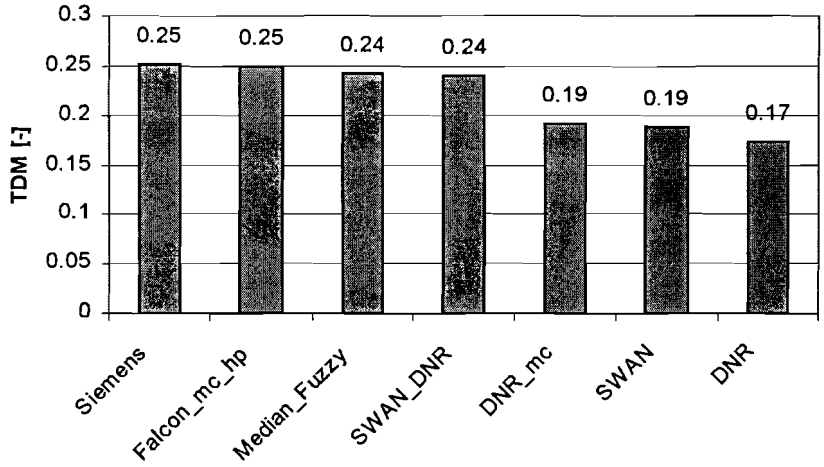

Figure B.15: *TDM measure evaluation for seven algorithms on 26 dB PSNR sequences*

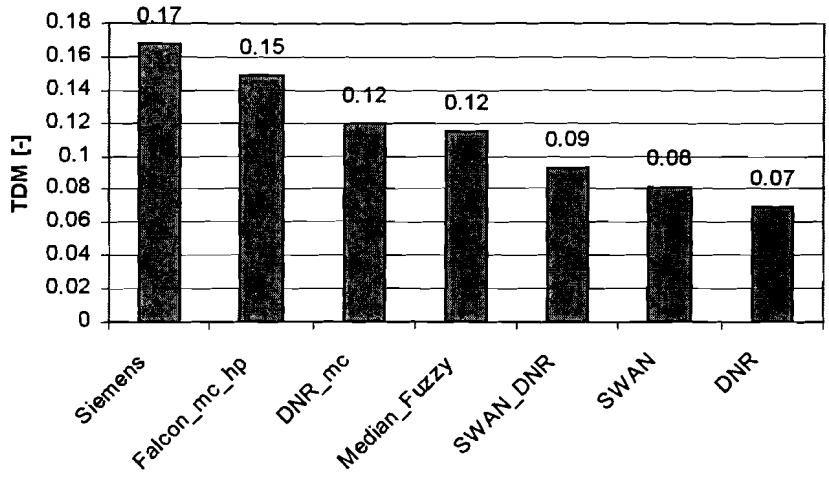

Figure B.16: *TDM measure evaluation for seven algorithms on 32 dB PSNR sequences*

In Figure B.17 and Figure B.18, the *Mdis* measure results are given. The algorithms DNR_mc and Siemens perform the best for the 26 dB *PSNR* noise degradation, while the Median_Fuzzy algorithm performs the worst. The DNR and Siemens algorithms perform the best for 32 dB *PSNR* noise degradation and Falcon_mc_hp performs the worst for this noise degradation.

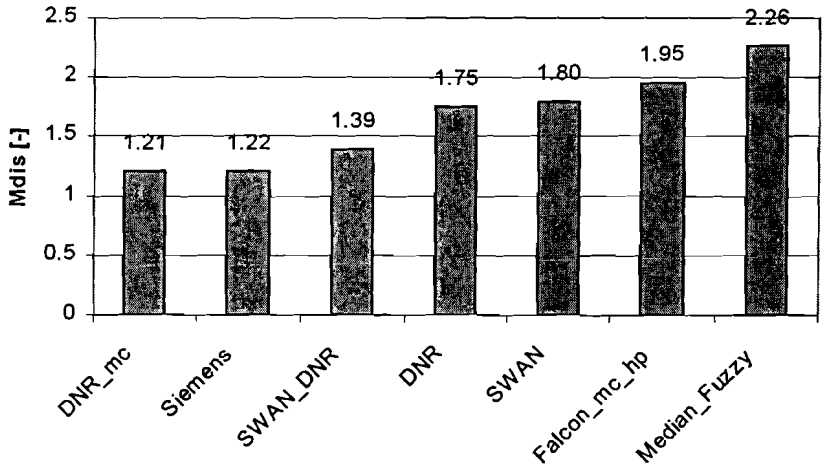

Figure B.17: *Mdis measure evaluation for seven algorithms on 26 dB PSNR sequences*

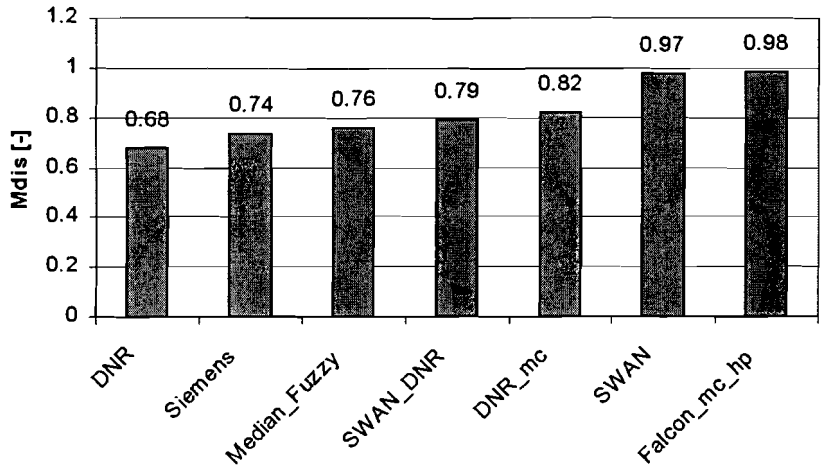

Figure B.18: *Mdis measure evaluation for seven algorithms on 32 dB PSNR sequences*

Appendix C

Subjective measurement results

C.1 Experiment 1

In Figure C.1, the five algorithms of the first test are ranked by the quality perception. The figure takes both noise degradations into account. The Siemens algorithm performed the best and the Median_Fuzzy performed the worst.

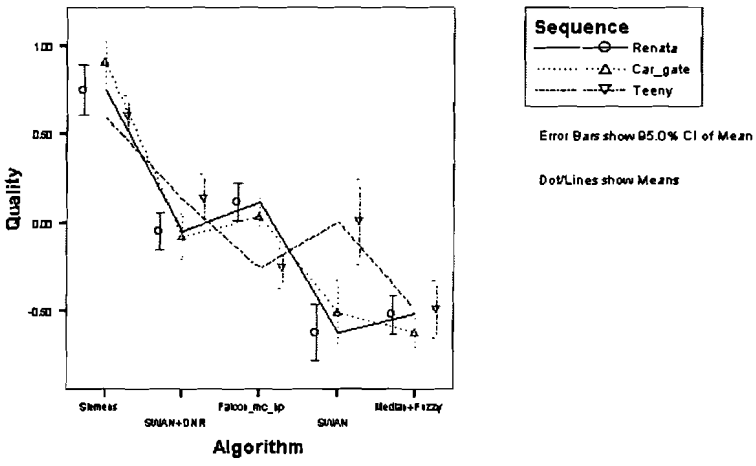

Figure C.1: *Quality ranking of five noise reduction algorithms of test 1*

In Figure C.2, the five algorithms of the first test are ranked by the noise removal performance.

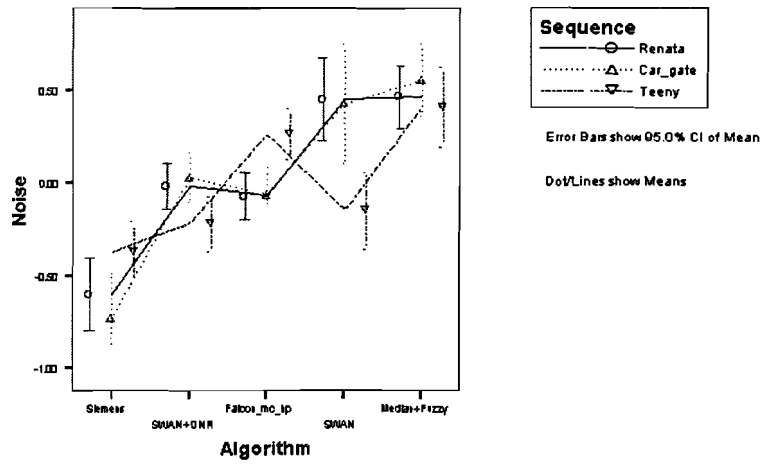

Figure C.2: *Noise ranking of five noise reduction algorithms of test 1*

In Figure C.3, the quality perception of the algorithms in experiment 1 is shown for sequences of 26 dB *PSNR*. Siemens has the best performing on the field of quality. The Median_Fuzzy algorithm gives the worst quality.

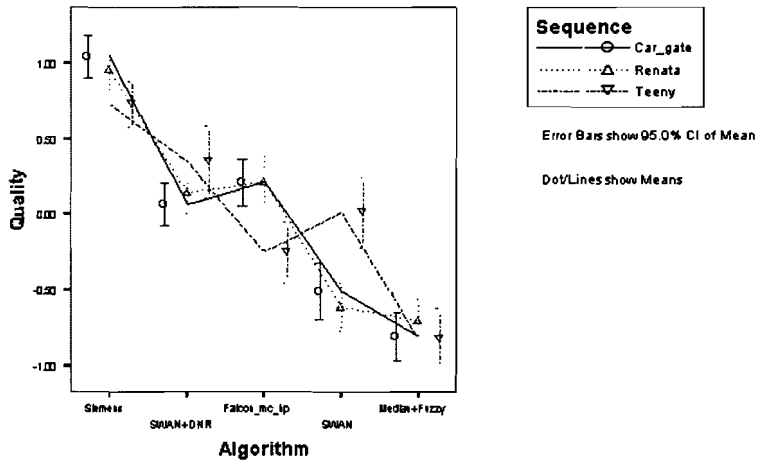

Figure C.3: *Quality ranking of five noise reduction algorithms of experiment 1 with 26 dB PSNR sequences*

In Figure C.4, the Noise scores of the algorithms in experiment 1 is shown with 26 dB *PSNR* sequences.

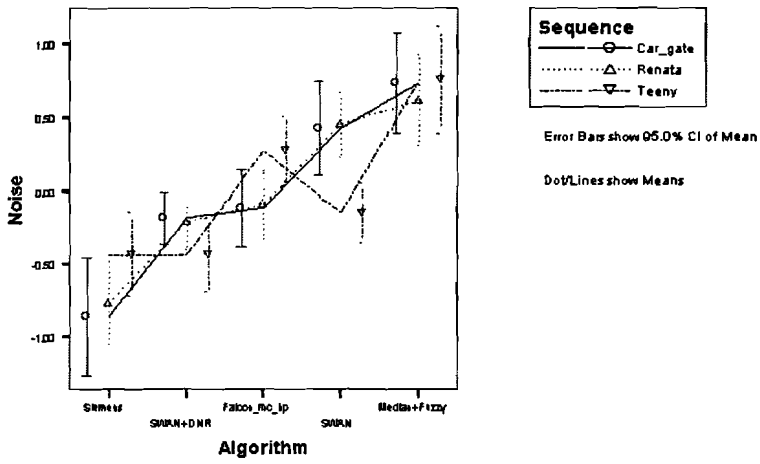

Figure C.4: *Noise ranking of five noise reduction algorithms of experiment 1 with 26 dB PSNR sequences*

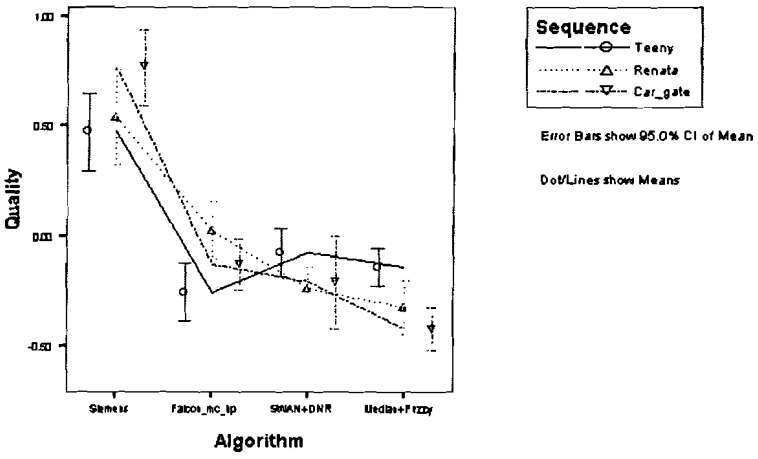

Figure C.5: *Quality ranking of four noise reduction algorithms of experiment 1 with 32 dB PSNR sequences*

In Figure C.5, the quality perception of the algorithms in experiment 1 is shown with 32 dB *PSNR* sequences.
In Figure C.6, the noise removal performance of the algorithms in experiment 1 is shown with 32 dB *PSNR* sequences. The Siemens algorithm removes the most noise, while the Median_Fuzzy algorithm performs the worst at noise removal ability.

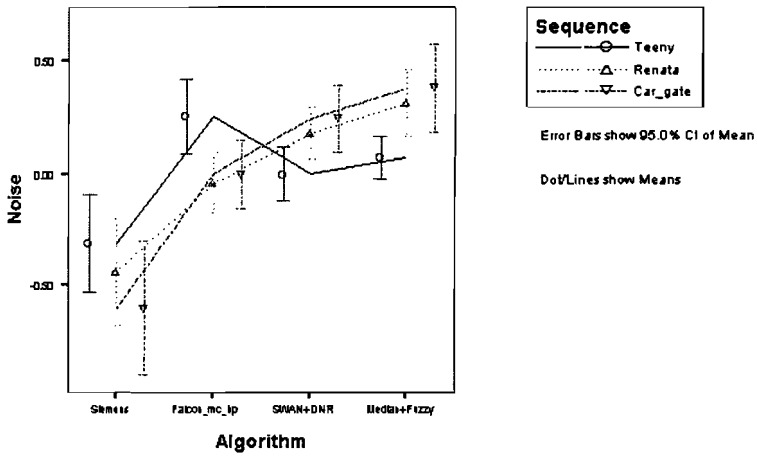

Figure C.6: *Noise ranking of four noise reduction algorithms of experiment 1 with 32 dB PSNR sequences*

C.2 Experiment 2

In Figure C.7, the five algorithms of the second test are ranked by Quality score.

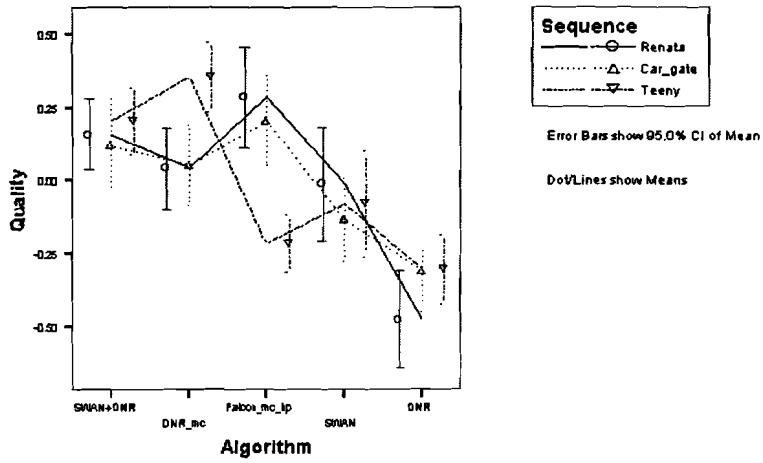

Figure C.7: *Quality ranking of five noise reduction algorithms of test 2*

In Figure C.8, the five algorithms of the second experiment are ranked by the Noise score.

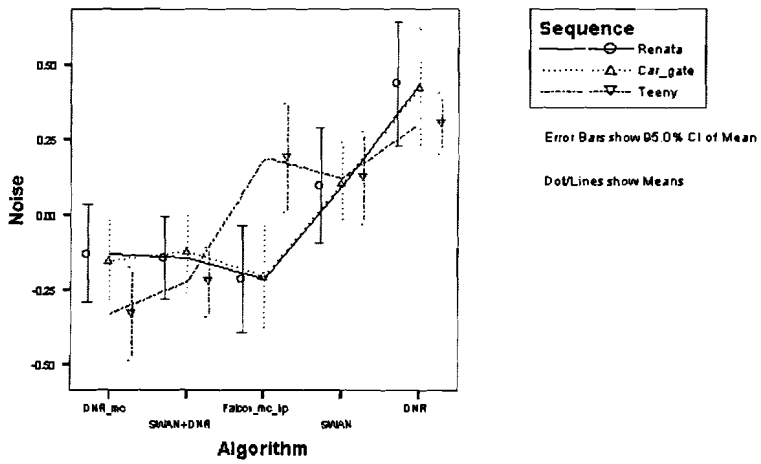

Figure C.8: *Noise ranking of five noise reduction algorithms of experiment 2*

In Figure C.9, the quality perception of the algorithms in experiment 2 is shown with 26 dB *PSNR* sequences.

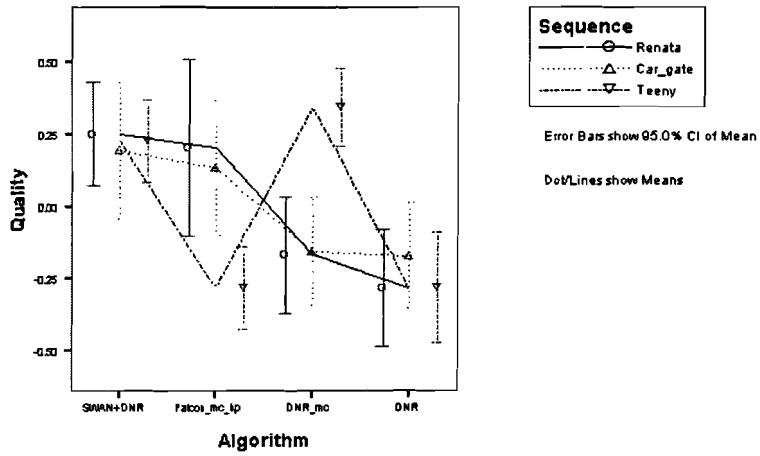

Figure C.9: *Quality ranking of four noise reduction algorithms of experiment 2 with 26 dB PSNR sequences*

In Figure C.10, the noise removal performance of the algorithms in experiment 2 is shown with 26 dB *PSNR* sequences. In Figure C.11 the quality perception of the algorithms in experiment 2 is shown for 32 dB *PSNR* sequences.

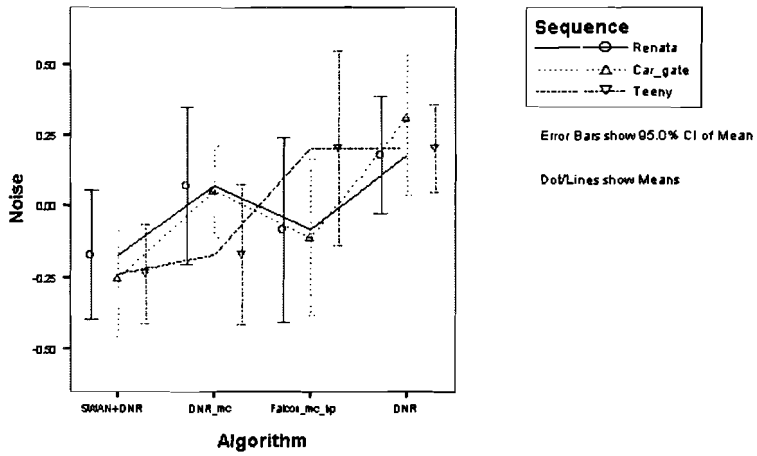

Figure C.10: *Noise ranking of four noise reduction algorithms of experiment 2 with 26 dB PSNR sequences*

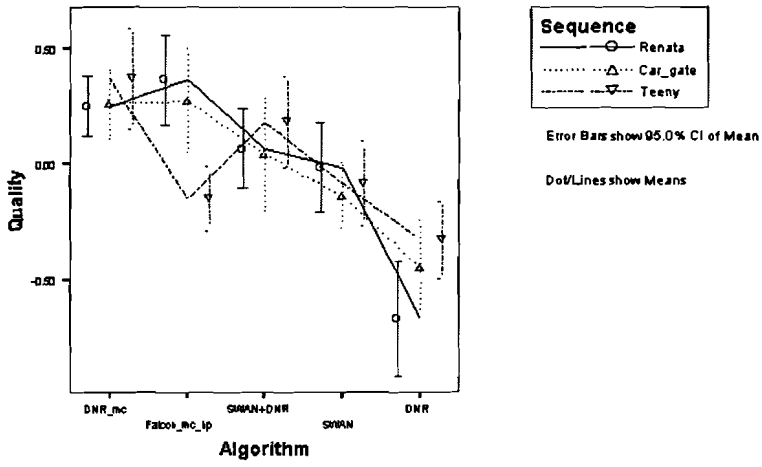

Figure C.11: *Quality ranking of five noise reduction algorithms of experiment 2 with 32 dB PSNR sequences*

In Figure C.12 the noise removal performance of the algorithms in experiment 2 is shown with 32 dB *PSNR* sequences.

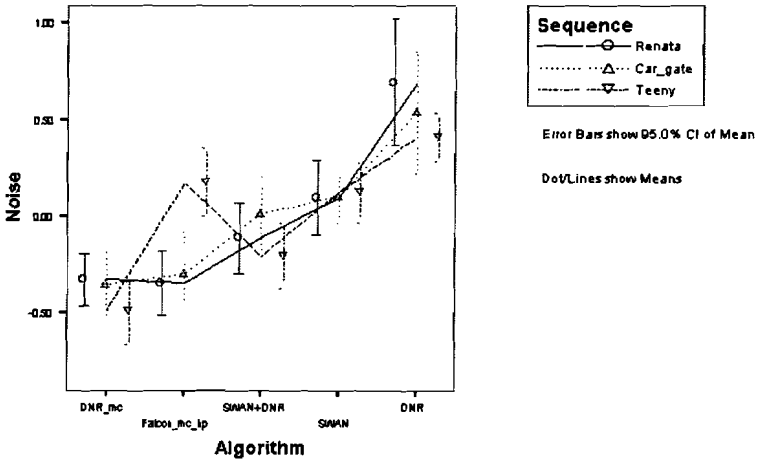

Figure C.12: *Noise ranking of five noise reduction algorithms of experiment 2 with 32 dB PSNR sequences*

Appendix D

Settings of the noise reduction algorithms

Table D.1: *Settings of the noise reduction algorithms for 26 dB PSNR sequences*

Filter	Spatial setting	Temporal setting
Falcon_mc_hp	n.a.	11
Siemens	n.a.	H=2, L=8
SWAN + DNR	12	8
Median + Fuzzy	n.a.	4
SWAN	15	n.a.
DNR	n.a.	5
DNR_mc	n.a.	8

Table D.2: *Settings of the noise reduction algorithms for 32 dB PSNR sequences*

Filter	Spatial setting	Temporal setting
Falcon_mc_hp	n.a.	8
Siemens	n.a.	H=2, L=6
SWAN + DNR	5	5
Median + Fuzzy	n.a.	3
SWAN	7	n.a.
DNR	n.a.	3
DNR_mc	n.a.	6

Appendix E

Data conversion differential to individual scores

D is the differential score of the difference between two algorithms, as directly obtained from the voting of the observers in the two experiments. The columns contain the scores per observer per sequence. S is the row matrix with individual scores for each algorithm. X is the transistion matrix, such that:

$$D = X * S \tag{E.1}$$

Then, S can be obtained from D and X, by [21]:

$$S = (X^T * S)^{-1} * X^T * D \tag{E.2}$$