3-DIMENSIONAL MEDIAN-BASED ALGORITHMS IN IMAGE SEQUENCE PROCESSING

A THESIS
SUBMITTED TO THE DEPARTMENT OF ELECTRICAL
AND ELECTRONICS ENGINEERING
AND THE INSTITUTE OF ENGINEERING AND SCIENCES
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

By
Münire Bilge ALP
September 1990

3-DIMENSIONAL MEDIAN-BASED ALGORITHMS IN IMAGE SEQUENCE PROCESSING

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND

ELECTRONICS ENGINEERING

AND THE INSTITUTE OF ENGINEERING AND SCIENCES

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

By

Münire Bilge Alp

September 1990

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Levent Onural(Principal Advisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assist. Prof. Dr. Enis Çetin

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

EnderAyanoglu

Assist. Prof. Dr. Ender Ayanoğlu

Approved for the Institute of Engineering and Sciences:

M. Baray

Prof. Dr. Mehmet Baray
Director of Institute of Engineering and Sciences

ABSTRACT

3-DIMENSIONAL MEDIAN-BASED ALGORITHMS IN IMAGE SEQUENCE PROCESSING

Münire Bilge Alp
M.S. in Electrical and Electronics Engineering
Supervisor: Assoc. Prof. Dr. Levent Onural
September 1990

This thesis introduces new 3-dimensional median-based algorithms to be used in two of the main research areas in image sequence processing: image sequence enhancement and image sequence coding. Two new nonlinear filters are developed in the field of image sequence enhancement. The motion performances and the output statistics of these filters are evaluated. The simulations show that the filters improve the image quality to a large extent compared to other examples from the literature. The second field addressed is image sequence coding. A new 3-dimensional median-based coding and decoding method is developed for stationary images with the aim of good slow motion performance. All the algorithms developed are simulated on real image sequences using a video sequencer.

ÖZET

GÖRÜNTÜ DİZİSİ İŞLEMEDE 3-BOYUTLU MEDYAN-BAZLI ALGORİTMALAR

Münire Bilge Alp
Elektrik ve Elektronik Mühendisliği Bölümü Yüksek Lisans
Tez Yöneticisi: Assoc. Prof. Dr. Levent Onural
Eylül 1990

Bu tez görüntü dizisi işlemenin iki ana dalında kullanılmak üzere yeni 3-boyutlu medyan-bazlı algoritmalar tanıtmaktadır. Görüntü dizisi iyileştirme alanında iki yeni, doğrusal olmayan süzgeç geliştirilmiştir. Bu süzgeçlerin hareket başarımları ve çıkış istatistikleri hesaplanmıştır. Benzetimler geliştirilen süzgeçlerin bugüne kadar yayınlanan örnekleriyle karşılaştırıldığında görüntü niteliğini büyük ölçüde iyileştirdiğini göstermektedir. Üzerinde çalışılan ikinci alan görüntü dizisi kodlamadır. Durağan görüntüler için yeni bir 3-boyutlu medyan-bazlı kodlayıcı ve kod çözücü geliştirilmiştir. Bu tasarımda amaç aynı zamanda iyi bir yavaş hareket başarımı elde etmektir. Geliştirilen bütün algoritmaların benzetimleri video dizici kullanılarak gerçek görüntü dizileri üzerinde yapılmıştır.

ACKNOWLEDGMENT

First of all, I would like to thank Assoc. Prof. Levent Onural for giving me the chance to work at Tampere University of Technology in Finland where this thesis was realized. I am indebted to him for encouraging me to experience a new and different working environment.

I want to express my deep gratitude to Prof. Yrjö Neuvo for his supervision and invaluable advice throughout my studies in Finland. I wish I had enough time to use all the bright ideas he has given me. I hope I will be able to continue cooperating with him.

I also owe special thanks to Petri Jarske without whose encouragement I would have never been able to complete this thesis. He has always been an invaluable friend to me in days of despair. I am indebted to Petri Haavisto, Janne Juhola, Vesa Lunden, and Tiina Jarske for the support they have given me and to all the people working in Signal Processing Laboratory in Tampere University of Technology for providing a friendly and efficient atmosphere. Finally, I want to thank my friends in Turkey and my family for the encouragement and love they have given me even from so far away.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

Image processing has been an active area of research for many years. Much effort has been given to the extension of 1-dimensional algorithms to two dimensions, taking the properties of images into account. However, many applications in image processing require the processing of 3-dimensional signals, namely image sequences. TV applications, target tracking, robot navigation, dynamic monitoring of industrial processes, study of cell motion by microcinematography, highway traffic monitoring, and video transmission are only a few examples where the signal to be processed is 3-dimensional, the third dimension being time. It has been shown in many cases that 1-dimensional algorithms do not produce optimum results in image processing. In other words, while processing images, their 2-dimensional nature has to be taken into account. Likewise, in processing image sequences, 1- or 2-dimensional algorithms do not yield optimum results. Although similar in some respects, the extension of 2-dimensional algorithms to 3-dimensional signal processing is not straightforward. The motion content of the image sequence requires the time dimension to be approached in a different manner.

Temporal filters have been developed to make use of the information in the time dimension in many image processing problems like image coding [1], noise filtering [2], and scan rate upconversion [3]. However, temporal filters usually blur the moving parts of the image sequence, resulting in poor image quality.

It is known that 2-dimensional spatial processing gives better results in moving parts, whereas temporal processing gives better results in still parts of the image sequence. This observation leads to the development of adaptive algorithms that require motion estimation or motion compensation to obtain acceptable image quality. Some examples are adaptive encoders [4], adaptive scan rate converters [5], adaptive color processors [6], and adaptive noise filters [7]. However, motion estimation and motion compensation are critical processes which increase the complexity and the cost of processing. Therefore, it is highly desirable to have 3-dimensional filters which would be insensitive to motion in image sequences.

In this thesis, two of the main research areas in image sequence processing are addressed. These are image sequence enhancement and image sequence coding. Two new 3-dimensional nonlinear filtering algorithms are introduced for noise filtering of image sequences. The motion performances of these filters are analyzed and their statistical properties are obtained. They are compared with the other 2- and 3-dimensional algorithms from the literature [2,7,9]. The algorithms are simulated using a video sequencer (DVSR 100), and examples of their application to real image sequences are presented.

Good results have been reported for the use of adaptive encoders in image sequence coding [4,29,30]. However, there is a problem with these adaptive encoders: their performance relies heavily on the motion detection and estimation algorithm. False decisions cause visible disturbances in the image quality. To prevent this, the algorithms used for still parts of the image should be able to perform fairly well in slow motion and vice versa. In this thesis, a new 3-dimensional coding and decoding method is developed for stationary images with the aim of good slow motion performance. The algorithm may be used as part of an adaptive encoder, or on its own.

All the filtering algorithms that are introduced in both image sequence enhancement and image sequence coding are based on the median operation. In developing these structures, the detail preserving property of the median operation has been made use of. In image processing, median-type operations are known for their edge-preserving capability. Since slow motion in an image sequence can be considered as an edge in the time dimension, good results can be expected with 3-dimensional median-based algorithms. 3-Dimensional median-based processing is a new research area in image processing where so far very few results have been published [9].

The organization of this thesis is as follows. In Section II, the median operation and its several extensions are defined. Some properties of the median operation are given and the tools necessary for the theoretical analysis are introduced. Section III provides both the theoretical and the practical results of the research carried on image sequence enhancement. After a brief background, the newly developed filters are described along with the other 2- and 3-dimensional algorithms. Further, these algorithms are analyzed for comparison purposes. In addition to the theoretical analysis, simulation results on DVSR image sequencer are presented. In Section IV, the algorithm developed for image sequence coding is defined and its performance is examined. Finally, Section V gives some conclusions and possible future work.

Chapter 2

MEDIAN OPERATION IN SIGNAL PROCESSING

In 1974 Tukey used the median operation for smoothing statistical data for the first time [8]. Since then the median operation has been widely used, especially in image processing tasks. Many generalizations and modifications of the median operation have been defined [10,11,12,13]. Median-based operations are known for their capability of following the abrupt changes in the signal, thus reducing blurring to a large extent. It has been shown via psychophysical experiments that a distorted image with sharp edges can be subjectively more pleasing than the original [14]. In image processing tasks, median filters preserve edges and high frequency details in the image, resulting in improved image quality. Active research and development is still going on for the theoretical analysis, as well as the applications and implementations of the median-based operations.

2.1 Definitions

Standard median filters are a subclass of the nonlinear filters called stack filters. They perform a windowed filtering operation where a window of fixed size moves over the input signal. The operation is nonlinear: at each position of the window, the median value of the data within the window is taken as the output [15]. For an odd window size of $N = 2k + 1$, the median of the input signal

$(x_1, \ldots, x_N)$ is defined as the $(k+1)$st largest value in the sorted sequence. So, if

$$x_{(1)} \leq x_{(2)} \leq \cdots \leq x_{(k+1)} \leq \cdots \leq x_{(2k+1)}$$

is the sorted input sequence, the output of the standard median filter is given as

$$y(n) = MED[x_1, \ldots, x_N] = x_{(k+1)}. \tag{2.1}$$

For even window size, the median can be taken as the average of the two middle samples in the sorted sequence [16]. However, in most cases the window size is fixed to be an odd integer.

The median of N input samples $(x_1, \ldots, x_N)$ can also be defined as the value that minimizes the mean absolute error, i.e.,

$$\sum_{i=1}^{N} |x_{med} - x_i| \leq \sum_{i=1}^{N} |y - x_i| \quad , \text{ for all } y \, . \tag{2.2}$$

If the window size is odd then the median is unique and is always one of the input samples. If the window size is even then there can be an infinite number of possible values that minimize the mean absolute error.

The median of a biexponentially distributed input sequence gives the maximum likelihood estimate of the mean of the distribution. If $(x_1, \ldots, x_N)$ are random variables with a probability density function

$$f(x) = \frac{\alpha}{2} e^{-\alpha|x-\beta|} \tag{2.3}$$

where $\alpha > 0$ is a scalar and β is the mean, the maximum likelihood estimate of β is given by $MED[x_1, \ldots, x_N]$. This can be easily proved by taking the logarithm of the likelihood function

$$L(\beta) = \left(\frac{\alpha}{2}\right)^N \prod_{i=1}^{N} e^{-\alpha|x_i-\beta|} \tag{2.4}$$

and maximizing it with respect to β. Median operation has several properties that make it suitable for image processing tasks. First, its response to an impulse is zero, implying that it is very effective in attenuating impulsive noise.

Second, its step response is a step, implying that it preserves abrupt changes in the signal, therefore reduces blurring. Finally, since the output of a median filter is always one of the input samples, there are signals that pass through the median filter unchanged. These are known as the root signals of the filter. Since median filters are nonlinear and do not have a passband in the sense of linear filters, these root signals may be considered as the passband of the filter.

Many generalizations and modifications of the median operation have been introduced [10,11,13]. Since we will later use them, we now describe one of these modifications: the weighted median filters. In weighted median filters, each sample x_i within the window is associated with a corresponding weight W_i. Usually W_i's are restricted to be positive integers, and $\sum_i W_i$ is odd, but the definition can easily be extended to non-integer weights. For positive integer weights, each sample is duplicated as many times as its weight and the median of the overall sequence is taken as the output [17]. The notation $< W_1, \ldots, W_N >$ will be used to show the weighted median filters. This can be illustrated by a simple example.

Example: The output of the filter $< 1, 1, 3, 1, 1 >$ is obtained as

$$y = MED[x_1, x_2, x_3, x_3, x_3, x_4, x_5],$$

$$= MED[x_1, x_2, 3 \diamond x_3, x_4, x_5].$$

Here $\diamond$ shows the weighting operation. If this filter is applied to an input sequence $\underline{x} = (3, 2, 4, 5, 1)$ the output will be

$$y = MED[3, 2, 4, 4, 4, 5, 1] = 4.$$

An equivalent definition of the weighted median filter is given as the value y that minimizes the sum,

$$\phi(y) = \sum_{i=1}^{N} W_i |x_i - y| \,. \tag{2.5}$$

It can be shown that both definitions are equivalent when W_i's are restricted to be positive integers [24]. Finally, it should be noted that standard median filters are a subclass of weighted median filters where all the weights are fixed as unity.

2.2 Threshold Decomposition and Stacking Property

Both the median filters and the weighted median filters are a subclass of stack filters. Therefore they satisfy the two basic properties that define a stack filter, i.e., the stacking property and the threshold decomposition property. These properties are essential tools for the theoretical analysis of the median-based filters, and are described below.

Two signals $\underline{x} = (x_1, \ldots, x_N)$ and $\underline{y} = (y_1, \ldots, y_N)$ "stack" if $x_i \geq y_i$ for each $i \in \{1, \ldots, N\}$. This is denoted as $\underline{x} \geq \underline{y}$. A filter $S(.)$ is said to possess the *stacking property* if and only if

$$S(\underline{x}) \geq S(\underline{y}) \text{ whenever } \underline{x} \geq \underline{y}. \tag{2.6}$$

Stacking property is a consistency rule which guarantees that the order of the input signals will not be changed by filtering.

Threshold decomposition is used to decompose an M-valued signal into M-1 binary signals. Given an M-valued signal $\underline{X} = (X_1, \ldots, X_N)$, the M-1 binary signals can be obtained as follows:

$$x_i^m = \begin{cases} 1 & , \ X_i \geq m \\ 0 & , \ \text{otherwise} \end{cases} \quad , \quad m = 1, \ldots, M-1 \ . \tag{2.7}$$

The signal $\underline{X}$ can be expressed as the sum of its binary decompositions, i.e.,

$$\underline{X} = \sum_{m=1}^{M-1} \underline{x}_i^m \ . \tag{2.8}$$

Note that $x_i^s \leq x_i^t$ for each $i \in \{1, \ldots, N\}$ for $s \geq t$, i.e., the binary signals obtained by the threshold decomposition of $\underline{X}$ form a stack of zeros on top of a stack of ones.

If the output of a filter can be obtained by first threshold decomposing the input signal to M-1 levels, then filtering the signals at each threshold level in the binary domain, and then summing up the outputs at each level, the filter is said to possess the threshold decomposition (THD) property. This can be illustrated by a simple example shown in Figure 2.1.

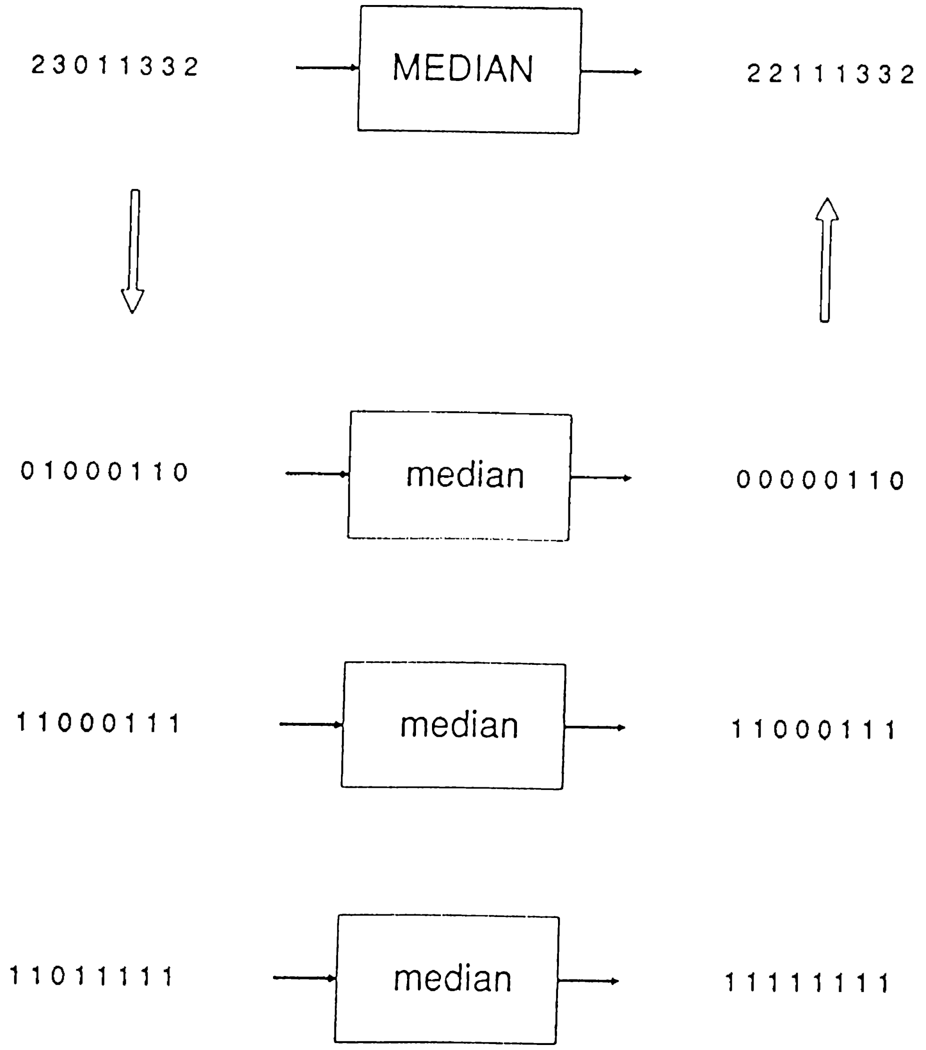


Figure 2.1 Threshold decomposition of 3-point standard median filter. In the binary domain, the median operation reduces to the application of a positive Boolean function on the input variables.

Threshold decomposition is a very useful tool which reduces the analysis of M-valued signals to binary signals. The results obtained in the binary domain can then be generalized to integer domain taking each level into account. This property together with the stacking property has been largely used since their development by Fitch *et al.* [18].

Chapter 3

IMAGE SEQUENCE ENHANCEMENT

3.1 General

The enhancement of noisy images has been extensively studied in the literature [19]. However, very little has been reported on the enhancement of image sequences. The extension to the third dimension has two major improvements over 2-dimensional algorithms. First, it gives a significant freedom to the designer by making various approaches possible. Second, the results obtained via 3-dimensional processing are far better than 2-dimensional processing since the information present in time is used. The major reason of the limited success obtained by 3-dimensional processing of image sequences is the insufficiency of many existing algorithms which deal with the motion in the sequence.

3-Dimensional linear FIR, IIR, and Kalman filters have been developed for the enhancement of image sequences [20,21]. These have been found to blur sharp edges as in the case of linear processing of 2-dimensional images in addition to the blurring in the moving areas. Since median-based filters are proved to be better than linear filters in the preservation of sharp edges and high frequency details, it is natural to expect the same improvement in 3-dimensional processing. In fact, the standard median filter has already been found to preserve the motion better than linear filters even in the straightforward application to time dimension [7]. However, it still requires motion detection and motion compensation to obtain good image quality. Adaptive filtering has been applied in the area of medical imaging where the enhancement of image sequences is of critical importance in spite of the increase in cost and complexity [22].

The first example of 3-dimensional median-based nonlinear filters developed with the aim of preserving motion has been given by Arce *et al.* [9]. These filters have been quite successful in the preservation of motion, but they have rather poor noise attenuation. As will be seen, substantial improvement over these algorithms has been obtained by the filters introduced in this thesis.

3.2 Filtering Structures

In this section, two new median-based 3-dimensional filtering structures will be introduced and their recursive versions will be defined. There are very few examples in the literature of this kind. For this reason two that are first introduced by Arce *et al.* will also be presented. Finally, the 2-dimensional algorithms that are developed and used for comparison purposes will be described.

3.2.1 3-Dimensional Planar Filter (P3D)

The first 3-dimensional algorithm is based on the multilevel median structure introduced in [23]. The structure is shown in Figure 3.1. It consists of four standard median filters. Each of the 5-point median operations in the first level operate on a different plane of the 3-dimensional image sequence, i.e., on the x-y, x-t, and y-t planes. This is the reason why the filter is called the 3-dimensional planar filter. For a discrete spatio-temporal image sequence given by $\{I(x,y,t) : x,y,t \in Z\}$ where Z is the set of integer numbers, the output of the 3-dimensional planar filter is defined as follows.

Definition 1:

The three first level filters are

$$\begin{aligned} m_{xy}(x,y,t) &= MED[I(x+r,y,t), I(x,y+r,t), I(x,y,t)]\ ,\ r=\pm 1\ ; \\ m_{xt}(x,y,t) &= MED[I(x+r,y,t), I(x,y,t+r), I(x,y,t)]\ ,\ r=\pm 1\ ; \\ m_{yt}(x,y,t) &= MED[I(x,y+r,t), I(x,y,t+r), I(x,y,t)]\ ,\ r=\pm 1\ , \end{aligned} \tag{3.1}$$

where r takes both +1 and -1 values, i.e., each standard median filter in the first level has five input variables, and the final output, $f_{P3D}\Big(I(x,y,t)\Big)$, is given by

$$\begin{aligned} Y(x,y,t) &= f_{P3D}\Big(I(x,y,t)\Big) \\ &= MED\Big[m_{xy}(x,y,t), m_{xt}(x,y,t), m_{yt}(x,y,t)\Big]\ . \end{aligned} \tag{3.2}$$

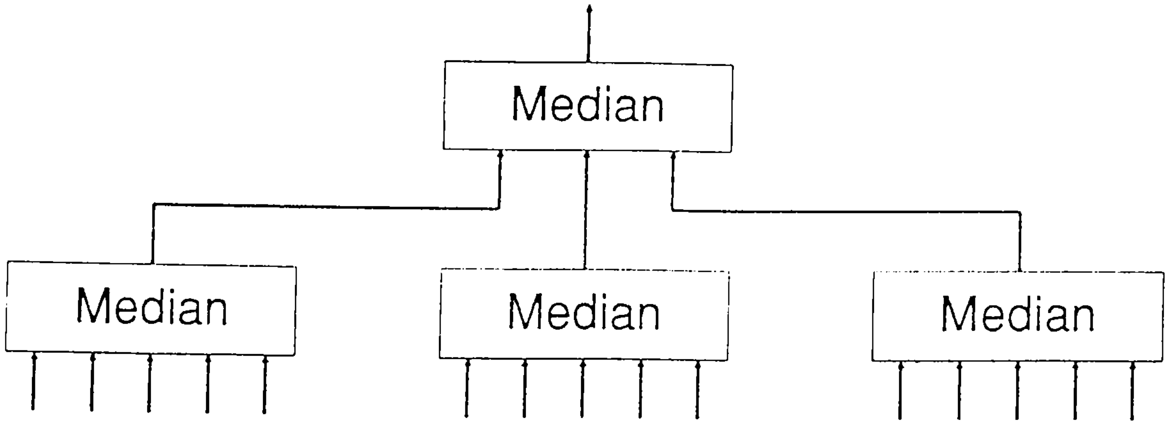


Figure 3.1 The multilevel structure for the 3-D planar filter (P3D).

Note that, when the image sequence is static, the consecutive frames are identical and thus, $I(x,y,t-1) = I(x,y,t) = I(x,y,t+1)$. In this case, the output sequence is equal to the input sequence, resulting in perfect reconstruction. The filter preserves all high frequency details of static image sequences.

Usually the recursive versions of median-based filters have higher noise attenuations. In this thesis, the recursive version of P3D is also developed. It is denoted by P3DR and is defined as follows.

Definition 2:

The three first level filters of P3DR are

$$\begin{aligned} m_{xy}(x,y,t) &= MED\Big[Y(x-1,y,t), Y(x,y-1,t), I(x,y,t), \\ &\qquad I(x+1,y,t), I(x,y+1,t)\Big] \; ; \\ m_{xt}(x,y,t) &= MED\Big[Y(x-1,y,t), Y(x,y,t-1), I(x,y,t), \\ &\qquad I(x+1,y,t), I(x,y,t+1)\Big] \; ; \\ m_{yt}(x,y,t) &= MED\Big[Y(x,y-1,t), Y(x,y,t-1), I(x,y,t), \\ &\qquad I(x,y+1,t), I(x,y,t+1)\Big] \; , \end{aligned} \tag{3.3}$$

and the final output, $f_{P3DR}\Big(I(x,y,t)\Big)$, is given by

$$Y(x,y,t) = MED\Big[m_{xy}(x,y,t), m_{xt}(x,y,t), m_{yt}(x,y,t)\Big] \; . \tag{3.4}$$

3.2.2 3-Dimensional Multilevel Filter (ML3D)

The second filter developed is based on the preservation of different features in the first level of the multilevel structure. The first level consists of two 7-point median filters each preserving different features of the input image. The multilevel structure is shown in Figure 3.2. For a spatio-temporal image sequence $\{I(x,y,t) : x,y,t \in Z\}$ where Z is the set of integers, the filter operation can be formulated as follows.

Definition 3:

The first level 7-point medians are

$$\begin{aligned} m_+(x,y,t) &= MED[I(x+r,y,t), I(x,y+r,t), \\ &\qquad I(x,y,t+r), I(x,y,t)]\ ,\ r = \pm 1; \\ m_\times(x,y,t) &= MED[I(x+r,y+r,t), I(x+r,y-r,t), \\ &\qquad I(x,y,t+r), I(x,y,t)]\ ,\ r = \pm 1, \end{aligned} \tag{3.5}$$

and the final output, $f_{ML3D}\Big(I(x,y,t)\Big)$, is

$$\begin{aligned} Y(x,y,t) &= f_{ML3D}\Big(I(x,y,t)\Big) \\ &= MED[m_+(x,y,t), m_\times(x,y,t), I(x,y,t)]\ . \end{aligned} \tag{3.6}$$

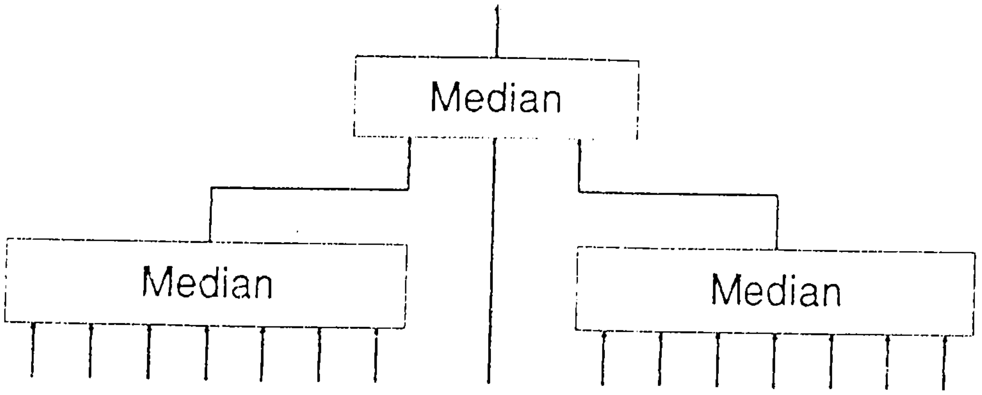


Figure 3.2 The multilevel structure for the 3-D multilevel filter (ML3D).

For comparison purposes, the first level outputs are analyzed separately. The first one (m_+) preserves plus-shaped features and is called PL3D. The second one ($m_\times$) preserves cross-shaped features and is called CR3D. It is possible to define the recursive versions of these filters, denoted by PL3DR, CR3DR, and ML3DR respectively, as follows.

Definition 4:

The first level 7-point medians are

$$\begin{aligned} m_+(x,y,t) &= MED\Big[Y(x-1,y,t), Y(x,y-1,t), Y(x,y,t-1), \\ &\qquad I(x,y,t), I(x+1,y,t), I(x,y+1,t), I(x,y,t+1)\Big] \; ; \\ m_\times(x,y,t) &= MED\Big[Y(x-1,y-1,t), Y(x+1,y-1,t), Y(x,y,t-1), \\ &\qquad I(x,y,t), I(x-1,y+1,t), I(x+1,y+1,t), I(x,y,t+1)\Big] \; , \end{aligned} \tag{3.7}$$

and the output, $f_{ML3DR}\Big(I(x,y,t)\Big)$, is

$$Y(x,y,t) = MED\Big[m_+(x,y,t), m_\times(x,y,t), I(x,y,t)\Big] \; . \tag{3.8}$$

3.2.3 Unidirectional (UNI3D) and Bidirectional (BI3D) Multistage Filters

In [9], Arce *et al.* introduced two types of multilevel median-based filters, i.e., unidirectional and bidirectional multistage filters. These filters are defined and some simulation results are given in [9]. In this thesis, in addition to the simulations, the theoretical analysis will also be carried out for these filter structures under a specified mask and they will be compared with the newly developed algorithms. For the sake of completeness, the definitions of the filters will also be given here.

Consider a spatio-temporal input sequence $\{I(x,y,t) : x,y,t \in Z\}$ where Z is the set of integer numbers. The unidirectional subwindows, W_1, W_2, W_3, W_4, W_5, of a $(2N+1) \times (2N+1) \times (2N+1)$ cubic window are defined as

$$\begin{aligned} W_1[I(x,y,t)] &= \{I(x+r,y,t) \quad : -N \le r \le N\} \; , \\ W_2[I(x,y,t)] &= \{I(x+r,y+r,t) : -N \le r \le N\} \; , \\ W_3[I(x,y,t)] &= \{I(x,y+r,t) \quad : -N \le r \le N\} \; , \\ W_4[I(x,y,t)] &= \{I(x+r,y-r,t) : -N \le r \le N\} \; , \\ W_5[I(x,y,t)] &= \{I(x,y,t+r) \quad : -N \le r \le N\} \; . \end{aligned} \tag{3.9}$$

These masks can be seen in Figure 3.3 for $N = 1$. Let

$$z_i(x,y,t) = MED\Big[I(.) \in W_i[I(x,y,t)]\Big] \ , \ 1 \leq i \leq 5 \ . \tag{3.10}$$

Definition 5:

Using equations (3.9) and (3.10) , the output of the unidirectional multistage filter is given by

$$Y(x,y,t) = MED\Big[z_{max}(x,y,t), z_{min}(x,y,t), I(x,y,t)\Big] \ , \tag{3.11}$$

where

$$\begin{aligned} z_{max}(x,y,t) &= \max_{1\leq i\leq 5}\Big[z_i(x,y,t)\Big] \ , \\ z_{min}(x,y,t) &= \min_{1\leq i\leq 5}\Big[z_i(x,y,t)\Big] \ . \end{aligned} \tag{3.12}$$

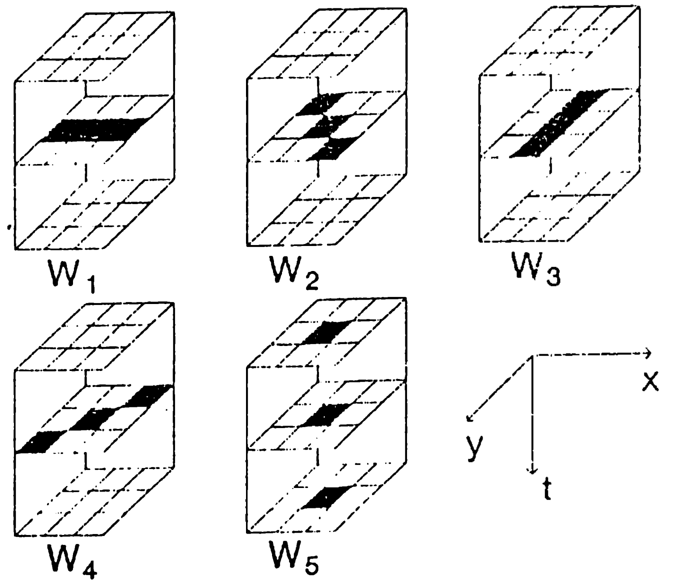


Figure 3.3 Unidirectional masks defined in (3.9) .

For bidirectional multistage filters the subwindows, $W_{(1,5)}$, $W_{(2,5)}$, $W_{(3,5)}$, $W_{(4,5)}$, of the cubic window are also of bidirectional type and are given as

$$W_{(i,5)}\Big[I(x,y,t)\Big] = W_i\Big[I(x,y,t)\Big] \bigcup W_5\Big[I(x,y,t)\Big] \ , \ 1 \leq i \leq 4 \ . \tag{3.13}$$

The bidirectional subwindows are shown in Figure 3.4 for a 3 × 3 × 3 cubic window. Let

$$z_{(i,5)}(x,y,t) = MED\Big[I(\cdot) \in W_{(i,5)}[I(x,y,t)]\Big] \ , \ 1 \leq i \leq 4 \ . \tag{3.14}$$

Definition 6:

Using equations (3.13) and (3.14) , the output of the bidirectional multistage filter is given by

$$Y^{+}(x,y,t) = MED\left[z^{+}_{max}(x,y,t), z^{+}_{min}(x,y,t), I(x,y,t)\right] , \qquad (3.15)$$

where

$$\begin{aligned} z^{+}_{max}(x,y,t) &= \max\nolimits_{1\leq i\leq 4}\left[z_{(i,5)}(x,y,t)\right] , \\ z^{+}_{min}(x,y,t) &= \min\nolimits_{1\leq i\leq 4}\left[z_{(i,5)}(x,y,t)\right] . \end{aligned} \qquad (3.16)$$

It is also possible to define the recursive unidirectional and bidirectional filters as usual.

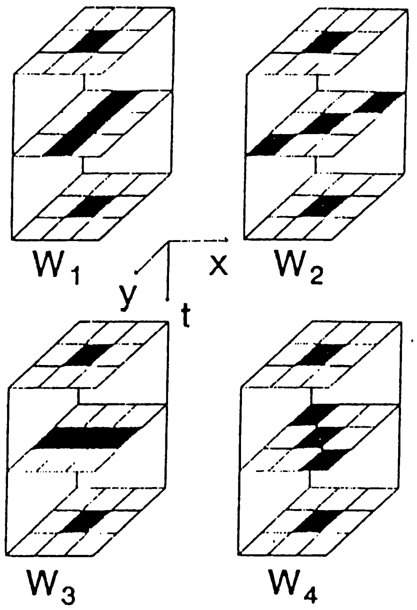


Figure 3.4 Bidirectional subwindows defined in (3.13) .

3.2.4 2-Dimensional Filters

Basically, three 2-dimensional filters have been used to make comparisons with the 3-dimensional algorithms that have been developed. The first one is the simple (+)-shaped 5-point median filter (MEDIAN5) given by

$$Y(x,y,t) = MED\left[I(x+r,y,t), I(x,y,t), I(x,y+r,t)\right] , \ r = \pm 1 , \qquad (3.17)$$

where r takes both of the values +1 and -1. The second nonlinear filter that is developed is the 2-dimensional counterpart of ML3D. Instead of taking pixels from the previous and next frames, a weight of three is given to the center pixel $I(x, y, t)$. This filter is called multilevel weighted median filter (MLW2D) and is formulated as follows.

Definition 7:

The two first level filters are

$$\begin{aligned} m_+(x,y,t) &= MED\Big[I(x+r,y,t), I(x,y+r,t), 3 \diamond I(x,y,t)\Big] , \\ m_\times(x,y,t) &= MED\Big[I(x+r,y+r,t), I(x+r,y-r,t), 3 \diamond I(x,y,t)\Big] , \end{aligned} \tag{3.18}$$

where $r = \pm 1$, and the final output is

$$Y(x,y,t) = MED\Big[m_+(x,y,t), m_\times(x,y,t), I(x,y,t)\Big] . \tag{3.19}$$

Finally, the last 2-dimensional filter used for comparison is a linear averaging filter (LAVE) in a 3×3 square window given as

$$\begin{aligned} Y(x,y,t) = \Big[I(x-1,y-1,t) &+ I(x,y-1,t) + I(x+1,y-1,t) \\ + I(x-1,y,t) &+ I(x,y,t) + I(x+1,y,t) \\ + I(x-1,y+1,t) &+ I(x,y+1,t) + I(x+1,y+1,t)\Big]\Big/9 . \end{aligned} \tag{3.20}$$

3.3 Derivation of the Boolean Functions

A Boolean function is positive if and only if it contains no complements of its input variables in its minimum sum of products (MSP) form. Each positive Boolean function (PBF) has a unique MSP form [24]. It has been shown that PBF's have the stacking property, i.e., each PBF represents a stack filter [25]. Since multilevel median filters belong to the class of stack filters, there is a PBF corresponding to each of the filters defined in Section 3.2. These PBF's are used in the analysis of the filters in the binary domain. The results can then be extended to multi-valued signals by the threshold decomposition property.

The PBF corresponding to a stack filter can be found by listing all combinations of the input variables having value one such that the output of the filter will also be one. This expression can then be simplified to obtain the MSP form. For multilevel filters it may be complicated to find all possible combinations when the number of input variables increases. To overcome this difficulty, the PBF's for 3- and 5-point median operations given below are used in the derivations.

$$MED[x_1, x_2, x_3] = x_1x_2 + x_1x_3 + x_2x_3 \ . \tag{3.21}$$

$$\begin{aligned} MED[x_1, x_2, x_3, x_4, x_5] =& x_1x_2x_3 + x_1x_2x_4 + x_1x_2x_5 + x_1x_3x_4 \\ &+ x_1x_3x_5 + x_1x_4x_5 + x_2x_3x_4 + x_2x_3x_5 \\ &+ x_2x_4x_5 + x_3x_4x_5 \ . \end{aligned} \tag{3.22}$$

To simplify the expressions, the notation given in Figure 3.5 will be used in the derivations and the analysis. In this notation, the subscript '0' stands for the previous frame $(t-1)$, the subscript '1' stands for the current frame (t), and the subscript '2' stands for the next frame $(t+1)$. For the current frame, the notation for the pixels within the given mask can be summarized as follows.

$$\begin{array}{lll} A_1 = I(x-1, y-1, t) & B_1 = I(x, y-1, t) & C_1 = I(x+1, y-1, t) \\ D_1 = I(x-1, y, t) & E_1 = I(x, y, t) & F_1 = I(x+1, y, t) \\ G_1 = I(x-1, y+1, t) & H_1 = I(x, y+1, t) & I_1 = I(x+1, y+1, t) \end{array}$$

Only the proof for Proposition 1 will be given here. The proofs for other propositions can be found in Appendix A.

Proposition 1:

The PBF corresponding to P3D is given by

$$\begin{aligned} f_{P3D}(E_0, B_1, D_1, E_1, F_1, H_1, E_2) =& E_0E_1E_2 + B_1E_1H_1 + D_1E_1F_1 \\ &+ (E_0 + E_2)(B_1D_1E_1 + B_1D_1H_1 + B_1E_1F_1 + B_1F_1H_1 \\ &+ E_1F_1H_1 + D_1E_1H_1 + B_1D_1F_1 + D_1F_1H_1) \\ &+ E_0E_2(B_1 + H_1)(D_1 + F_1) \ . \end{aligned} \tag{3.23}$$

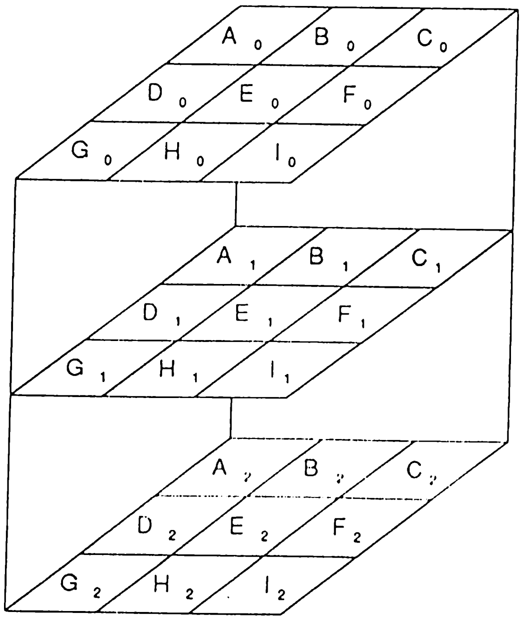


Figure 3.5 3 × 3 × 3 cubic mask representing 3 successive frames.

Proof :

Let Z_1, Z_2, Z_3 be the outputs of the three 5-point median filters on the first level of P3D. By equation (3.22) these can be expressed as

$$\begin{aligned}
Z_1 =& MED[B_1, D_1, E_1, F_1, H_1] \\
=& B_1D_1E_1 + B_1D_1F_1 + B_1D_1H_1 + B_1E_1F_1 + B_1E_1H_1 \\
&+ B_1F_1H_1 + D_1E_1F_1 + D_1E_1H_1 + D_1F_1H_1 + E_1F_1H_1 \ ; \\
Z_2 =& MED[E_0, D_1, E_1, F_1, E_2] \\
=& E_0D_1E_1 + E_0D_1F_1 + E_0D_1E_2 + E_0E_1F_1 + E_0E_1E_2 \\
&+ E_0F_1E_2 + D_1E_1F_1 + D_1E_1E_2 + D_1F_1E_2 + E_1F_1E_2 \ ; \\
Z_3 =& MED[E_0, B_1, E_1, H_1, E_2] \\
=& E_0B_1E_1 + E_0B_1H_1 + E_0B_1E_2 + E_0E_1H_1 + E_0E_1E_2 \\
&+ E_0H_1E_2 + B_1E_1H_1 + B_1E_1E_2 + B_1H_1E_2 + E_1H_1E_2 \ .
\end{aligned}$$

The output of P3D is given by the median of Z_1, Z_2, and Z_3.

$$\begin{aligned} f_{P3D}(E_0, B_1, D_1, E_1, F_1, H_1, E_2) &= MED[Z_1, Z_2, Z_3] \\ &= Z_1Z_2 \ + \ Z_1Z_3 \ + \ Z_2Z_3 \ . \end{aligned} \tag{3.24}$$

where

$$\begin{aligned} Z_1Z_2 =& E_0B_1D_1E_1 \ + \ E_0B_1D_1F_1 \ + \ E_0B_1E_1F_1 \ + \ E_0B_1D_1H_1E_2 \\ &+ E_0D_1E_1H_1 \ + \ E_0D_1F_1H_1 \ + \ E_0E_1F_1H_1 \ + \ E_0B_1E_1H_1E_2 \\ &+ B_1D_1E_1E_2 \ + \ B_1D_1F_1E_2 \ + \ B_1E_1F_1E_2 \ + \ E_0B_1F_1H_1E_2 \\ &+ D_1E_1H_1E_2 \ + \ D_1F_1H_1E_2 \ + \ E_1F_1H_1E_2 \ + \ D_1E_1F_1 \ ; \\ Z_1Z_3 =& E_0B_1D_1E_1 \ + \ E_0B_1D_1H_1 \ + \ E_0D_1E_1H_1 \ + \ E_0B_1D_1F_1E_2 \\ &+ E_0B_1E_1F_1 \ + \ E_0B_1H_1F_1 \ + \ E_0E_1F_1H_1 \ + \ E_0D_1E_1F_1E_2 \\ &+ B_1D_1E_1E_2 \ + \ B_1D_1H_1E_2 \ + \ D_1E_1H_1E_2 \ + \ E_0D_1H_1F_1E_2 \\ &+ B_1E_1F_1E_2 \ + \ B_1H_1F_1E_2 \ + \ E_1F_1H_1E_2 \ + \ B_1E_1H_1 \ ; \\ Z_2Z_3 =& E_0B_1D_1E_1 \ + \ E_0D_1E_1H_1 \ + \ E_0B_1D_1E_2 \ + \ E_0B_1D_1F_1H_1 \\ &+ E_0D_1H_1E_2 \ + \ E_0B_1E_1F_1 \ + \ E_0E_1F_1H_1 \ + \ E_0B_1F_1E_2 \\ &+ E_0F_1H_1E_2 \ + \ E_0E_1E_2 \ + \ B_1D_1E_1F_1H_1 \\ &+ B_1D_1E_1F_1E_2 \ + \ D_1E_1F_1H_1E_2 \ . \end{aligned}$$

Substituting the expressions given above for Z_1Z_2, Z_1Z_3, and Z_2Z_3 in (3.24) and making the simplifications using Boolean algebra results in the expression given in (3.23) .

Proposition 2:

The PBF's corresponding to the 7-point median filters, PL3D and CR3D, in the first level of 3-dimensional multilevel filter are

$$\begin{aligned} f_{PL3D}(E_0,B_1,D_1,E_1,F_1,H_1,E_2) &= B_1D_1F_1H_1 \\ &+E_0E_1E_2(B_1+D_1+F_1+H_1)+(E_0E_1+E_0E_2 \\ &+E_1E_2)(B_1D_1+B_1F_1+B_1H_1+D_1F_1+D_1H_1+F_1H_1) \\ &+(E_0+E_1+E_2)(B_1D_1F_1+B_1D_1H_1+B_1F_1H_1+D_1F_1H_1)\ ; \end{aligned} \tag{3.25}$$

$$\begin{aligned} f_{CR3D}(E_0,A_1,C_1,E_1,G_1,I_1,E_2) &= A_1C_1G_1I_1 \\ &+E_0E_1E_2(A_1+C_1+G_1+I_1)+(E_0E_1+E_0E_2 \\ &+E_1E_2)(A_1C_1+A_1G_1+A_1I_1+C_1G_1+C_1I_1+G_1I_1) \\ &+(E_0+E_1+E_2)(A_1C_1G_1+A_1C_1I_1+A_1G_1I_1+C_1G_1I_1)\ , \end{aligned} \tag{3.26}$$

and the PBF corresponding to ML3D is

$$\begin{aligned} f_{ML3D}(E_0,A_1,B_1,C_1,D_1,E_1,F_1,G_1,H_1,I_1,E_2) &= E_1(A_1C_1G_1 \\ &+A_1C_1I_1+A_1G_1I_1+C_1G_1I_1+B_1D_1F_1+B_1D_1H_1 \\ &+B_1F_1H_1+D_1F_1H_1)+(E_0+E_2)(A_1C_1G_1+A_1C_1I_1 \\ &+A_1G_1I_1+C_1G_1I_1)(B_1D_1F_1+B_1D_1H_1+B_1F_1H_1 \\ &+D_1F_1H_1)+E_1(E_0+E_2)(A_1C_1+A_1G_1+A_1I_1 \\ &+C_1G_1+C_1I_1+G_1I_1+B_1D_1+B_1F_1+B_1H_1+D_1F_1 \\ &+D_1H_1+F_1H_1)+E_0E_2(A_1C_1+A_1G_1+A_1I_1 \\ &+C_1G_1+C_1I_1+G_1I_1)(B_1D_1+B_1F_1+B_1H_1 \\ &+D_1F_1+D_1H_1+F_1H_1)+E_0E_1E_2(A_1+B_1+C_1+D_1 \\ &+F_1+G_1+H_1+I_1)+A_1B_1C_1D_1F_1G_1H_1I_1\ . \end{aligned} \tag{3.27}$$

Proposition 3:

The PBF corresponding to Arce's unidirectional multistage filter for a $3 \times 3 \times 3$ cubic window ($N = 1$) is given by

$$\begin{aligned} f_{UNI3D}(E_0,A_1,B_1,C_1,D_1,E_1,F_1,G_1,H_1,I_1,E_2) = E_1(A_1 + B_1 \\ +C_1 + D_1 + F_1 + G_1 + H_1 + I_1 + E_0 + E_2) \\ +A_1B_1C_1D_1F_1G_1H_1I_1E_0E_2 \,. \end{aligned} \tag{3.28}$$

Proposition 4:

The PBF corresponding to Arce's bidirectional multistage filter for a $3 \times 3 \times 3$ cubic window ($N = 1$) is given by

$$\begin{aligned} f_{BI3D}(E_0,A_1,B_1,C_1,D_1,E_1,F_1,G_1,H_1,I_1,E_2) = E_1(B_1H_1 + D_1F_1 \\ +A_1I_1 + C_1G_1) + (E_0 + E_2)A_1B_1C_1D_1F_1G_1H_1I_1 + E_0E_1E_2 \\ +E_1(E_0 + E_2)(A_1 + B_1 + C_1 + D_1 + F_1 + G_1 + H_1 + I_1) \\ +E_0E_2(B_1 + H_1)(D_1 + F_1)(A_1 + I_1)(C_1 + G_1) \,. \end{aligned} \tag{3.29}$$

Proposition 5:

The PBF's corresponding to 2-dimensional weighted median filters, PLW2D and CRW2D, in the first level of multilevel weighted median filter, MLW2D, are

$$f_{PLW2D}(B_1,D_1,E_1,F_1,H_1) = B_1D_1F_1H_1 + E_1(B_1 + D_1 + F_1 + H_1) \,; \tag{3.30}$$

$$f_{CRW2D}(A_1,C_1,E_1,G_1,I_1) = A_1C_1G_1I_1 + E_1(A_1 + C_1 + G_1 + I_1) \,, \tag{3.31}$$

and the PBF corresponding to multilevel weighted median filter, MLW2D, is

$$\begin{aligned} f_{MLW2D}(A_1,B_1,C_1,D_1,E_1,F_1,G_1,H_1,I_1) = A_1B_1C_1D_1F_1G_1H_1I_1 \\ + E_1(A_1 + B_1 + C_1 + D_1 + F_1 + G_1 + H_1 + I_1) \,. \end{aligned} \tag{3.32}$$

3.4 Some Observations on Root Signal Structures in Binary Domain

The Boolean functions obtained in Section 3.3 may be used to analyze the behaviour of the filters in binary domain. This section does not intend to give a complete root signal analysis of the filters described. Only some observations will be made on the root signal structures of these filters based on the corresponding Boolean expressions.

Observation 1:

The unidirectional multistage filter (UNI3D) introduced by Arce *et al.* is equivalent to the following weighted median filter.

$$Y(x,y,t) = MED[E_0, A_1, B_1, C_1, D_1, 9 \diamond E_1, F_1, G_1, H_1, I_1, E_2] \, .$$

This observation follows directly from (3.28) , since this is the same expression for the PBF of the weighted median filter given above.

Observation 2:

The 2-dimensional multilevel median filter (MLW2D) given in (3.19) is equivalent to the following weighted median filter.

$$Y(x,y,t) = MED[A_1, B_1, C_1, D_1, 7 \diamond E_1, F_1, G_1, H_1, I_1] \, .$$

The positive Boolean functions obtained for UNI3D and MLW2D show that these algorithms filter only single impulsive points within their masks, like a '0' in the middle of all 1's or a '1' in the middle of all 0's. In the integer domain this corresponds to the maximum and minimum points, i.e., the input pixel is changed only if it is an extremum point within the filtering mask. So, all image sequences that do not contain single impulses are roots of UNI3D and MLW2D.

The behaviour of 3-dimensional planar filter (P3D), 3-dimensional multilevel filter (ML3D), and bidirectional multistage filter (BI3D) can be analyzed in three separate cases according to the motion content of the image sequence. The first case corresponds to stationary sequences, the second one corresponds to slowly moving sequences where two successive frames out of three are equal, and the last one corresponds to fast moving sequences where each successive frame is different from the one before.

Case 1: $E_0 = E_1 = E_2$. The first case corresponds to stationary sequences. In this case the outputs of the filters can be expressed as follows.

$$E_0 = E_1 = E_2 \Rightarrow \begin{cases} f_{P3D}(\cdot) = E_1 \; ; \\ f_{BI3D}(\cdot) = E_1 \; ; \\ f_{ML3D}(\cdot) = A_1B_1C_1D_1F_1G_1H_1I_1 \\ \qquad\qquad + E_1(A_1 + B_1 + C_1 + D_1 \\ \qquad\qquad\qquad + F_1 + G_1 + H_1 + I_1). \end{cases} \tag{3.33}$$

Observation 3:

The above expressions show that both P3D and BI3D preserve all high frequency details in a stationary sequence, i.e., all stationary image sequences are root signals of P3D and BI3D. The 3-dimensional multilevel filter, ML3D, still eliminates an impulsive point even if it repeats in successive frames. As will be seen in Section 3.6, ML3D has the highest noise attenuation, which is expected.

Case 2: $E_0 = E_1 \neq E_2$ or $E_0 \neq E_1 = E_2$. In the second case, only two successive pixels out of three frames are equal. This may be considered as slow motion in a binary image sequence.

Observation 4:

In this case (case 2), the output of P3D reduces to

$$\begin{aligned} f_{P3D} = & B_1E_1H_1 + D_1E_1F_1 + B_1D_1E_1 + B_1D_1H_1 + B_1E_1F_1 \\ & + B_1F_1H_1 + E_1F_1H_1 + D_1E_1H_1 + B_1D_1F_1 + D_1F_1H_1 \; , \end{aligned} \tag{3.34}$$

which may be recognized as the 5-point median filter $MED[B_1, D_1, E_1, F_1, H_1]$. Since the filter reduces to a 2-dimensional algorithm, it is expected to preserve slow motion in the image sequence.

Observation 5:

The output of BI3D can be expressed as follows under slow motion.

$$\begin{aligned} f_{BI3D} =& A_1B_1C_1D_1F_1G_1H_1I_1 \ + \ E_1(A_1 + B_1 + C_1 \\ & + D_1 + F_1 + G_1 + H_1 + I_1)\,. \end{aligned} \tag{3.35}$$

This implies that bidirectional multistage filter attenuates only impulsive points within the 3 × 3 square mask under slow motion. Its noise attenuation is expected to be lower than that of P3D.

Observation 6:

The 3-dimensional multilevel filter, ML3D, preserves the input pixel, E_1, only if at least two other pixels in one of the substructures corresponding to $+$ or $\times -$ shaped features are equal to the input pixel. This implies that the filter preserves all lines of arbitrary width under slow motion.

Case 3: $E_0 \neq E_1 \neq E_2 (E_0 = E_2)$. In the case of fast motion, all successive pixels in three frames are different. In the binary domain this corresponds to oscillation in the time dimension.

Observation 7:

In this case, the following observation can be made on the output of P3D.

$$f_{P3D}(\cdot) = E_1 \iff B_1 = E_1 = H_1 \quad \text{or} \quad D_1 = E_1 = F_1\,. \tag{3.36}$$

This observation follows from the Boolean expression for the output of P3D under fast motion which can be expressed as

$$\begin{aligned} E_1 = 0 &\Rightarrow f_{P3D}(\cdot) = (B_1 + H_1)(D_1 + F_1)\,; \\ E_1 = 1 &\Rightarrow f_{P3D}(\cdot) = B_1H_1 \ + \ D_1F_1\,. \end{aligned} \tag{3.37}$$

The observation above implies that, under fast motion, the filter preserves vertical and horizontal lines of arbitrary width, and diagonal lines that are at least two pixels wide.

Observation 8:

In the case of fast motion (case 3), the bidirectional multistage filter preserves all lines of arbitrary width, i.e.,

$$f_{BI3D}(\cdot) = E_1 \iff \begin{array}{l} B_1 = E_1 = H_1 \quad \text{or} \quad D_1 = E_1 = H_1 \quad \text{or} \\ A_1 = E_1 = I_1 \quad \text{or} \quad C_1 = E_1 = G_1 \ . \end{array} \tag{3.38}$$

Observation 9:

Under fast motion (case 3), the 3-dimensional multilevel filter, ML3D, preserves the input pixel only if at least 3 other pixels in one of the substructures corresponding to $+$ or $\times -$ shaped features are equal to the current pixel, E_1. This implies that the filter preserves all features at least two pixels wide under fast motion. This reduction in resolution is not critical since the eye does not require high spatial resolution under fast motion.

3.5 Statistical Analysis

By using the Boolean expressions derived in Section 3.3, it is possible to express the output probability distribution functions in terms of the input distributions. An accurate statistical model for general, non-stationary sequences has not been developed yet. However, the noise attenuation of the filters can still be obtained for the homogeneous parts of the image where the problem is to estimate a constant signal in additive white noise. Along edges and under motion, the structural analysis should be used to evaluate the performance of the filter.

Let the input sequence, $I(x, y, t)$ be an independent, identically distributed (i.i.d.) discrete random field. The probability space of the input is given by

(Ω, F, P) where $\Omega = \{0, \ldots, M-1\}$ is the sample field, F is the event space, and P is the probability measure with the discrete distribution function $F(j), j \in \Omega$. The binary sequence obtained by threshold decomposing the input at level j can be expressed as

$$I^j(x,y,t) = \begin{cases} 1 \ , & I(x,y,t) \geq j \\ 0 \ , & \text{otherwise} \end{cases} \tag{3.39}$$

This binary sequence also forms an i.i.d. random field with sample space $\Omega_b = \{0,1\}$, and probability measure function,

$$\begin{aligned} Pr\{I^j = 0\} &= F(j-1) \ ; \\ Pr\{I^j = 1\} &= 1 - F(j-1) \ . \end{aligned} \tag{3.40}$$

Given the input distribution function $F(j)$, the output distribution functions of the filters defined in Section 3.2 can be derived using the Boolean expressions. The derivation will be given only for P3D here. The derivations for other filters can be found in detail in Appendix B.

Proposition 6:

The output distribution function of the 3-dimensional planar filter, P3D, is given in terms of the input probability distribution function $F(j)$ as

$$F_{P3D}(j) = F(j)^3[3 + 20F(j) - 57F(j)^2 + 49F(j)^3 - 14F(j)^4] \ , \tag{3.41}$$

and the output probability density function of P3D is given in terms of the input probability distribution $(F(j))$ and density $(f(j))$ functions as

$$f_{P3D}(j) = f(j)F(j)^2[9 + 80F(j) - 285F(j)^2 + 294F(j)^3 - 98F(j)^4] \ . \tag{3.42}$$

Proof :

Let $T(\cdot)$ be the threshold function such that

$$T_j\Big(I(x,y,t)\Big) = I^j(x,y,t) = \begin{cases} 1, & I(x,y,t) \geq j \\ 0, & \text{otherwise} \end{cases} \ ;$$

and

$$T_j\Big(Y(x,y,t)\Big) = Y^j(x,y,t) = \begin{cases} 1, & Y(x,y,t) \geq j \\ 0, & \text{otherwise} \end{cases} \ ,$$

where $I(x,y,t)$ is the M-valued input sequence and $Y(x,y,t)$ is the M-valued output sequence. For the sake of simplicity in the expressions, the following notation will be used in the proof.

$$E_0 = I^{j+1}(x,y,t) \qquad B_1 = I^{j+1}(x,y-1,t) \qquad F_1 = I^{j+1}(x+1,y,t)$$
$$E_1 = I^{j+1}(x,y,t-1) \qquad D_1 = I^{j+1}(x-1,y,t) \qquad H_1 = I^{j+1}(x,y+1,t)$$
$$E_2 = I^{j+1}(x,y,t+1)$$

Then the output at $(j+1)$st threshold level, $Y^{j+1}(x,y,t)$, can be expressed as given in (3.23) .

$$F_{P3D}(j) = Pr\{Y(x,y,t) \le j\} = Pr\{Y^{j+1}(x,y,t) = 0\} \tag{3.43}$$

Let

$$\begin{aligned}
P_1 &= Pr\{Y^{j+1}(x,y,t) = 0 | E_0 = E_1 = E_2 = 0\} \ ; \\
P_2 &= Pr\{Y^{j+1}(x,y,t) = 0 | E_0 = E_1 = E_2 = 1\} \ ; \\
P_3 &= Pr\{Y^{j+1}(x,y,t) = 0 | E_0 = E_1 = 0, E_2 = 1\} \\
&= Pr\{Y^{j+1}(x,y,t) = 0 | E_0 = 1, E_1 = E_2 = 0\} \ ; \\
P_4 &= Pr\{Y^{j+1}(x,y,t) = 0 | E_0 = E_1 = 1, E_2 = 0\} \\
&= Pr\{Y^{j+1}(x,y,t) = 0 | E_0 = 0, E_1 = E_2 = 1\} \ ; \\
P_5 &= Pr\{Y^{j+1}(x,y,t) = 0 | E_0 = E_2 = 1, E_1 = 0\} \ ; \\
P_6 &= Pr\{Y^{j+1}(x,y,t) = 0 | E_0 = E_2 = 0, E_1 = 1\} \ .
\end{aligned}$$

Then, by the total probability theorem, the output distribution function can be expressed as

$$\begin{aligned}
F_{P3D}(j) = &P_1 Pr\{E_0 = E_1 = E_2 = 0\} + P_2 Pr\{E_0 = E_1 = E_2 = 1\} \\
&+ P_3 Pr\{E_0 = E_1 = 0,\ E_2 = 1\} + P_3 Pr\{E_0 = 1,\ E_1 = E_2 = 0\} \\
&+ P_4 Pr\{E_0 = E_1 = 1,\ E_2 = 0\} + P_4 Pr\{E_0 = 0,\ E_1 = E_2 = 1\} \\
&+ P_5 Pr\{E_0 = E_2 = 1,\ E_1 = 0\} + P_6 Pr\{E_0 = E_2 = 0,\ E_1 = 1\} \ .
\end{aligned} \tag{3.44}$$

The probabilities defined above are obtained from the Boolean expression for P3D (3.23) .

$$P_1 = 1 \ ;$$
$$P_2 = 0 \ ;$$

$$\begin{aligned}
P_3 &= Pr\{B_1D_1H_1 + B_1F_1H_1 + B_1D_1F_1 + D_1F_1H_1 = 0\} \\
&= Pr\{\text{There are less than 3 ones among } B_1, D_1, F_1, H_1\} \\
&= \binom{4}{0}F(j)^4 + \binom{4}{1}F(j)^3(1-F(j)) + \binom{4}{2}F(j)^2(1-F(j))^2 \\
&= 6F(j)^2 - 8F(j)^3 + 3F(j)^4 \;\; ;
\end{aligned}$$

$$\begin{aligned}
P_4 &= Pr\{B_1D_1 + B_1F_1 + B_1H_1 + D_1F_1 + D_1H_1 + F_1H_1 = 0\} \\
&= Pr\{\text{There are less than 2 ones among } B_1, D_1, F_1, H_1\} \\
&= \binom{4}{0}F(j)^4 + \binom{4}{1}F(j)^3(1-F(j)) \\
&= 4F(j)^3 - 3F(j)^4 \;\; ;
\end{aligned}$$

$$\begin{aligned}
P_5 &= Pr\{(B_1 + H_1)(D_1 + F_1) = 0\} \\
&= Pr\{(B_1 + H_1) = 0 \;\text{ or }\; (D_1 + F_1) = 0\} \\
&= 1 - Pr\{(B_1 + H_1) = 1 \;\text{ and }\; (D_1 + F_1) = 1\} \\
&= 1 - \Big[Pr\{(B_1 + H_1) = 1\}\Big]^2 = 1 - \Big[Pr\{B_1 = 1 \;\text{ or }\; H_1 = 1\}\Big]^2 \\
&= 1 - \Big[1 - Pr\{B_1 = 0 \;\text{ and }\; H_1 = 0\}\Big]^2 \\
&= 1 - \Big[1 - F(j)^2\Big]^2 = 2F(j)^2 - F(j)^4 \;\; ;
\end{aligned}$$

$$\begin{aligned}
P_6 &= Pr\{B_1H_1 + D_1F_1 = 0\} \\
&= Pr\{B_1H_1 = 0 \;\text{ and }\; D_1F_1 = 0\} = \Big[Pr\{B_1 = 0 \;\text{ or }\; H_1 = 0\}\Big]^2 \\
&= \Big[1 - Pr\{B_1 = 1 \;\text{ and }\; H_1 = 1\}\Big]^2 = \Big[1 - (1 - F(j))^2\Big]^2 \\
&= 4F(j)^2 - 4F(j)^3 + F(j)^4 \;\; ;
\end{aligned}$$

Substituting these expressions in (3.44) results in

$$\begin{aligned}
F_{P3D}(j) =& F(j)^3 + 2F(j)^2\,(1 - F(j))\,\big(6F(j)^2 - 8F(j)^3 + 3F(j)^4\big) \\
&+ 2F(j)\,(1 - F(j))^2\,\big(4F(j)^3 - 3F(j)^4\big) \\
&+ F(j)\,(1 - F(j))^2\,\big(2F(j)^2 - F(j)^4\big) \\
&+ F(j)^2\,(1 - F(j))\,\big(4F(j)^2 - 4F(j)^3 + F(j)^4\big) \\
=& F(j)^3\,\big[3 + 20F(j) - 57F(j)^2 + 49F(j)^3 - 14F(j)^4\big] \;.
\end{aligned}$$

Finally, the output probability density function is obtained from the distribution function.

Proposition 7:

The output probability distribution function of the 3-dimensional multilevel filter, ML3D, and its substructures, PL3D and CR3D, are

$$F_{PL3D}(j) = F_{CR3D}(j) = F(j)^4 \left[35 - 84F(j) + 70F(j)^2 - 20F(j)^3\right] \; ; \tag{3.45}$$

$$\begin{aligned} F_{ML3D}(j) =& F(j)^4 \Big[40 - 106F(j) + 84F(j)^2 + 60F(j)^3 \\ & - 195F(j)^4 + 190F(j)^5 - 88F(j)^6 + 16F(j)^7\Big] \, , \end{aligned} \tag{3.46}$$

and the corresponding probability density functions are

$$f_{PL3D}(j) = f_{CR3D}(j) = 140f(j)F(j)^3 \left[1 - 3F(j) + 3F(j)^2 - F(j)^3\right] \; ; \tag{3.47}$$

$$\begin{aligned} f_{ML3D}(j) =& 2f(j)F(j)^3 \Big[80 - 265F(j) + 252F(j)^2 + 210F(j)^3 \\ & - 780F(j)^4 + 855F(j)^5 - 440F(j)^6 + 88F(j)^7\Big] \, . \end{aligned} \tag{3.48}$$

Proposition 8:

The output distribution function of the unidirectional multistage filter for the 3 × 3 × 3 cubic mask, UNI3D, is given as

$$F_{UNI3D}(j) = F(j)\left[1 - (1 - F(j))^{10}\right] \; + \; (1 - F(j))\,F(j)^{10} \, , \tag{3.49}$$

and the probability density function is given as

$$\begin{aligned} f_{UNI3D}(j) = f(j)\Big[& 1 - F(j)^{10} + 10F(j)^9\,(1 - F(j)) \\ & + 10F(j)\,(1 - F(j))^9 - (1 - F(j))^{10}\Big] \, . \end{aligned} \tag{3.50}$$

Proposition 9:

The output distribution function of the bidirectional multistage filter, BI3D, for the $3 \times 3 \times 3$ cubic mask is

$$\begin{aligned} F_{BI3D}(j) = F(j)^3 \Big[& 21 - 80F(j) + 166F(j)^2 - 224F(j)^3 + 202F(j)^4 \\ & 120F(j)^5 + 45F(j)^6 - 11F(j)^7 + 2F(j)^8 \Big] . \end{aligned} \tag{3.51}$$

and the probability density function is

$$\begin{aligned} f_{BI3D}(j) = f(j)F(j)^2 \Big[& 63 - 320F(j) + 830F(j)^2 - 1344F(j)^3 \\ & + 1414F(j)^4 - 960F(j)^5 + 405F(j)^6 - 110F(j)^7 + 22F(j)^8 \Big] . \end{aligned} \tag{3.52}$$

Proposition 10:

The output distribution function of the 2-dimensional multilevel weighted median filter, MLW2D, and its substructures, PLW2D and CRW2D, are

$$F_{PLW2D}(j) = F_{CRW2D}(j) = F(j)\left[1 - (1 - F(j))^4\right] + (1 - F(j))\,F(j)^4; \tag{3.53}$$

$$F_{MLW2D}(j) = F(j)\left[1 - (1 - F(j))^8\right] + (1 - F(j))\,F(j)^8 . \tag{3.54}$$

and the corresponding probability density functions are

$$\begin{aligned} f_{PLW2D}(j) = f_{CRW2D}(j) = f(j)\Big[& 1 - F(j)^4 + 4F(j)^3 (1 - F(j)) \\ & + 4F(j)(1 - F(j))^3 - (1 - F(j))^4 \Big] ; \end{aligned} \tag{3.55}$$

$$\begin{aligned} f_{MLW2D}(j) = f(j)\Big[& 1 - F(j)^8 + 8F(j)^7 (1 - F(j)) \\ & + F(j)(1 - F(j))^7 - (1 - F(j))^8 \Big] . \end{aligned} \tag{3.56}$$

Although the closed form formulas for various noise distributions like Gaussian, biexponential, and uniform noise are rather complicated, it is possible to make the statistical analysis by numerical methods. The probability distribution and density functions of the filters are plotted in Figures 3.6-3.11 for various noise types and filters. These graphs show the noise attenuation of the filters relative to one another.

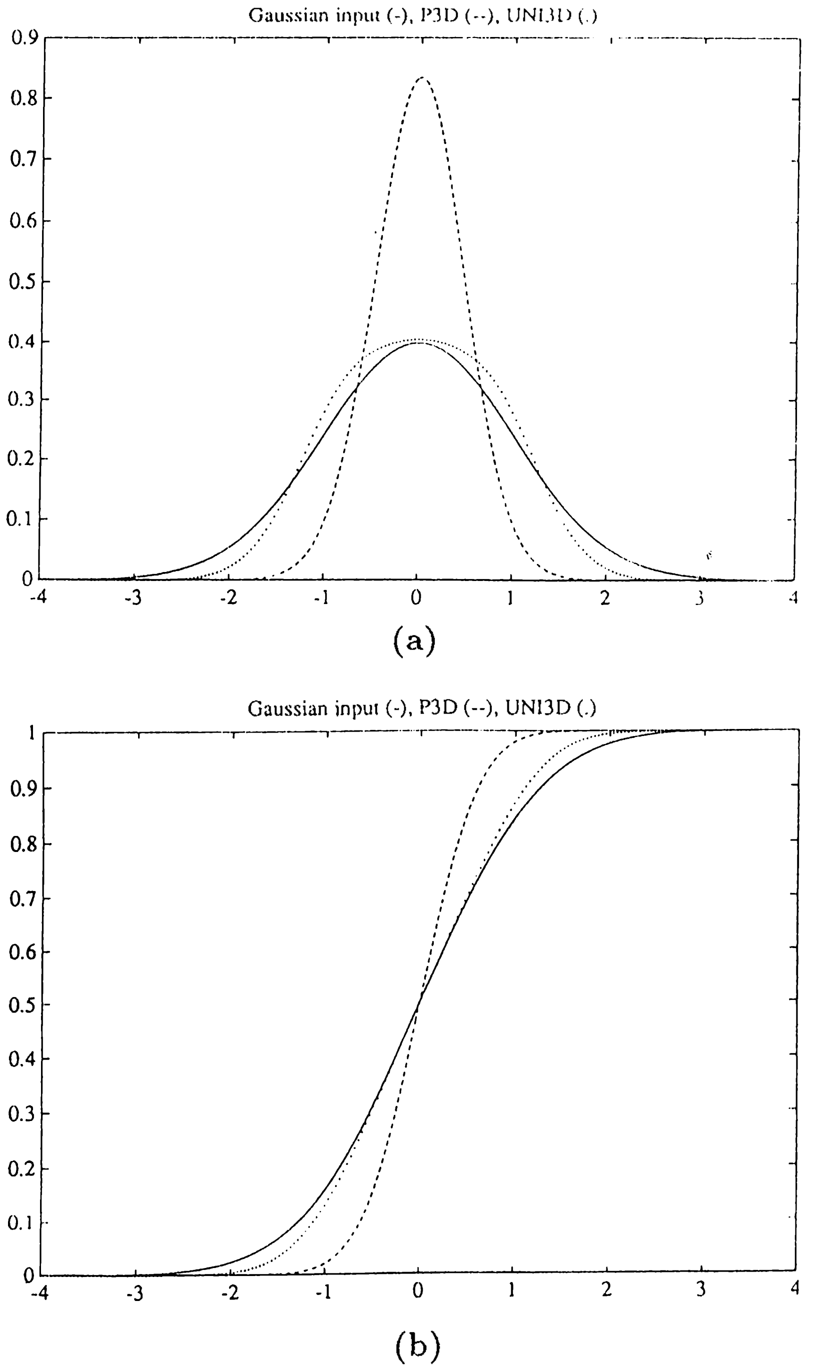
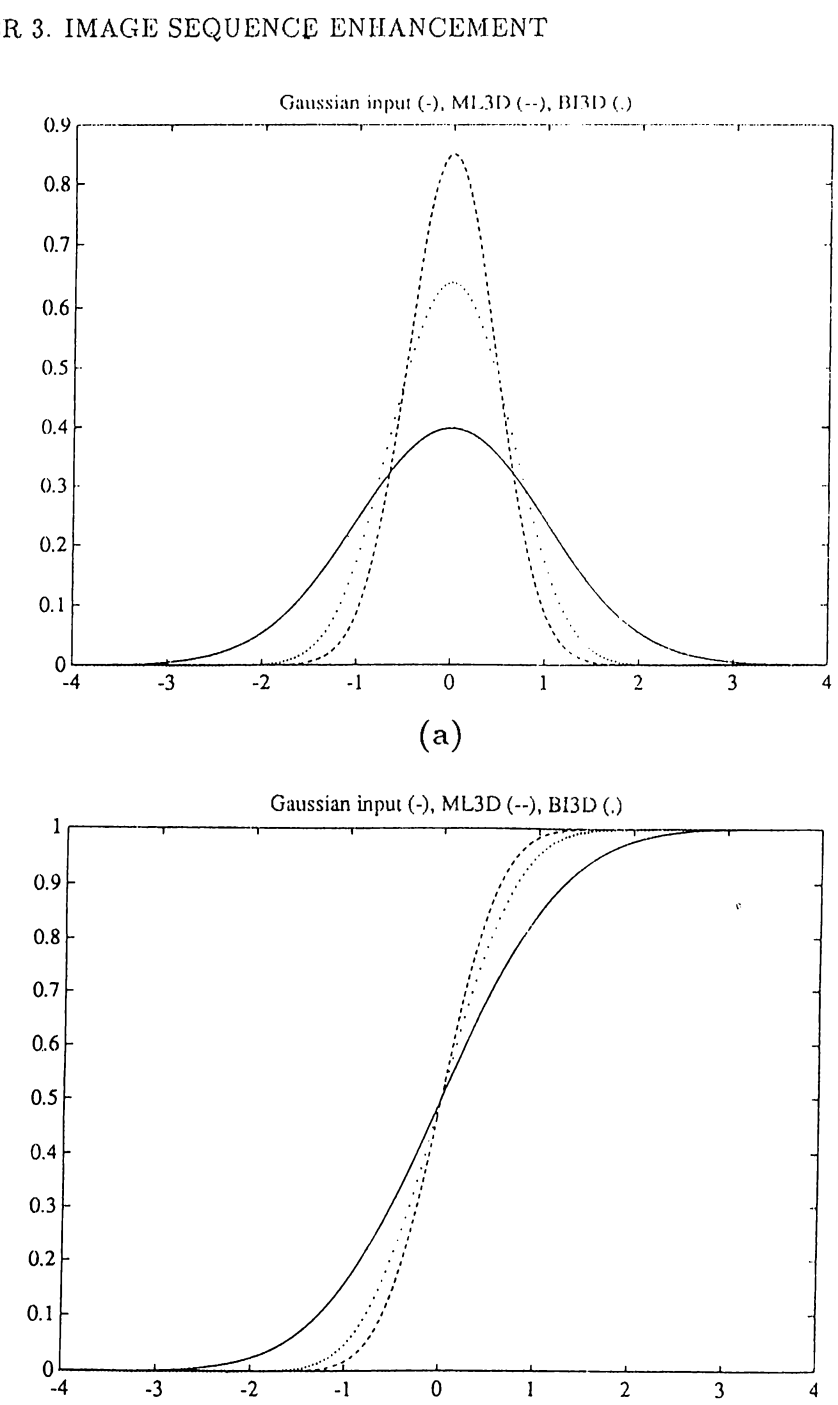


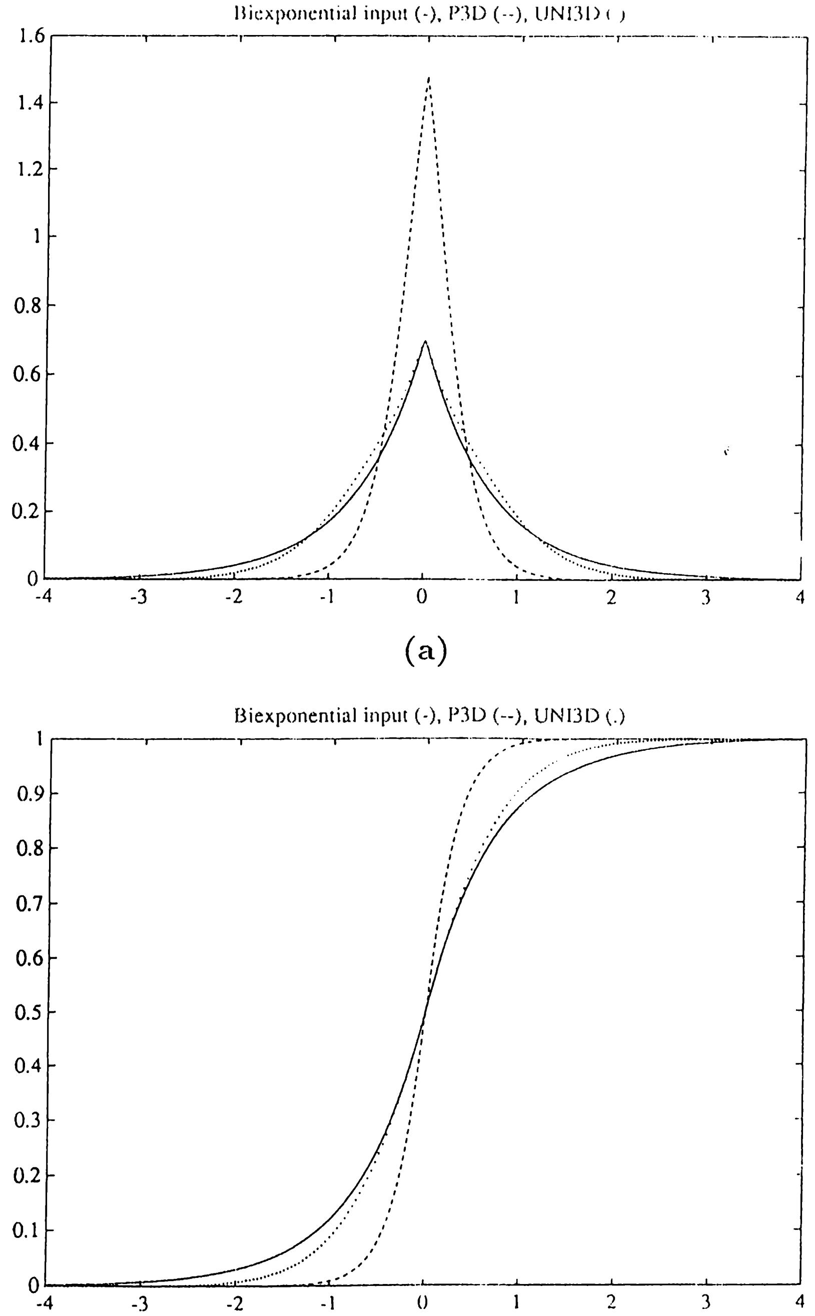
Figure 3.6 The output statistics of P3D, and UNI3D for zero mean, unit variance Gaussian noise, (a) the probability density functions, (b) the probability distribution functions.



(a)

(b)

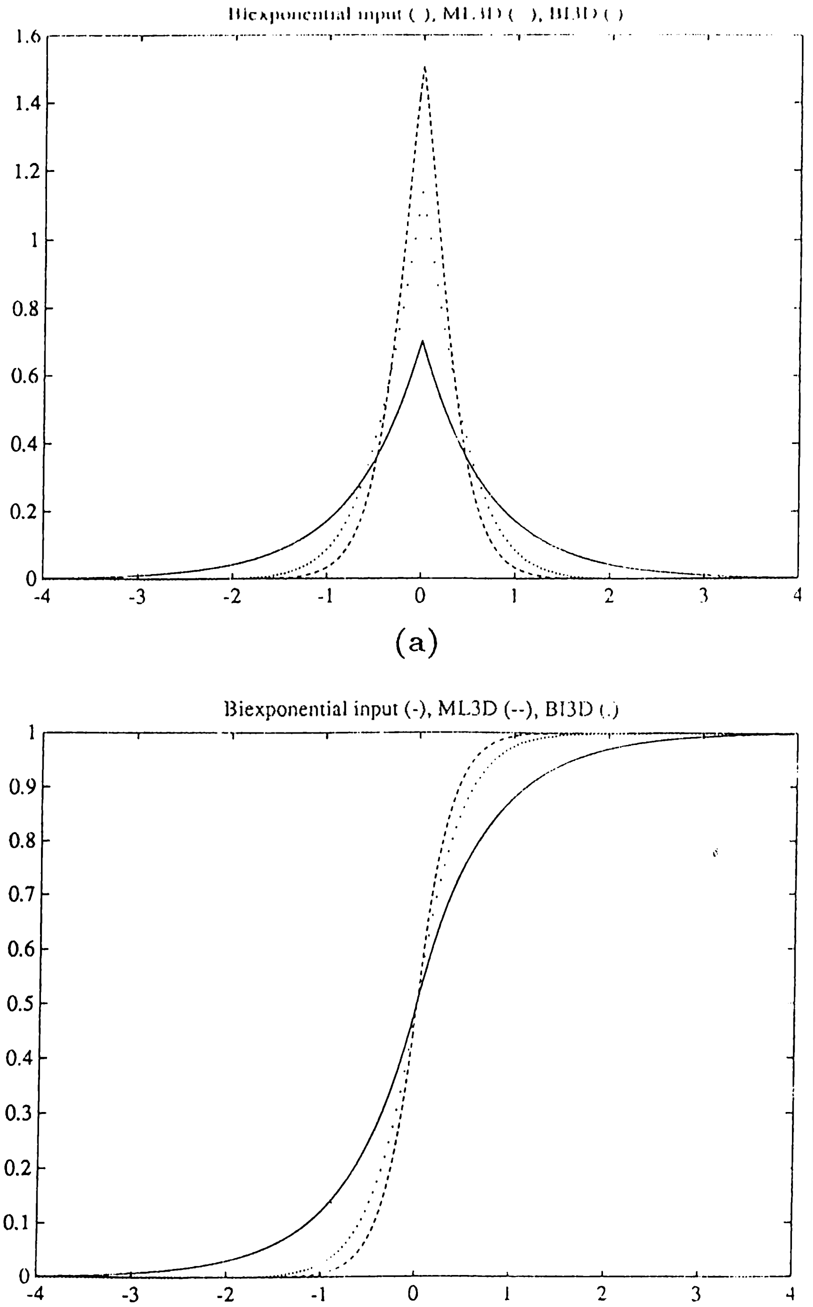
Figure 3.7 The output statistics of BI3D, and ML3D for zero mean, unit variance Gaussian noise, (a) the probability density functions, (b) the probability distribution functions.



(a)

(b)

Figure 3.8 The output statistics of P3D, and,UNI3D for zero mean, unit variance biexponential noise, (a) the probability density functions, (b) the probability distribution functions.



(a)

(b)

Figure 3.9 The output statistics of BI3D, and ML3D for zero mean, unit variance biexponential noise, (a) the probability density functions, (b) the probability distribution functions.

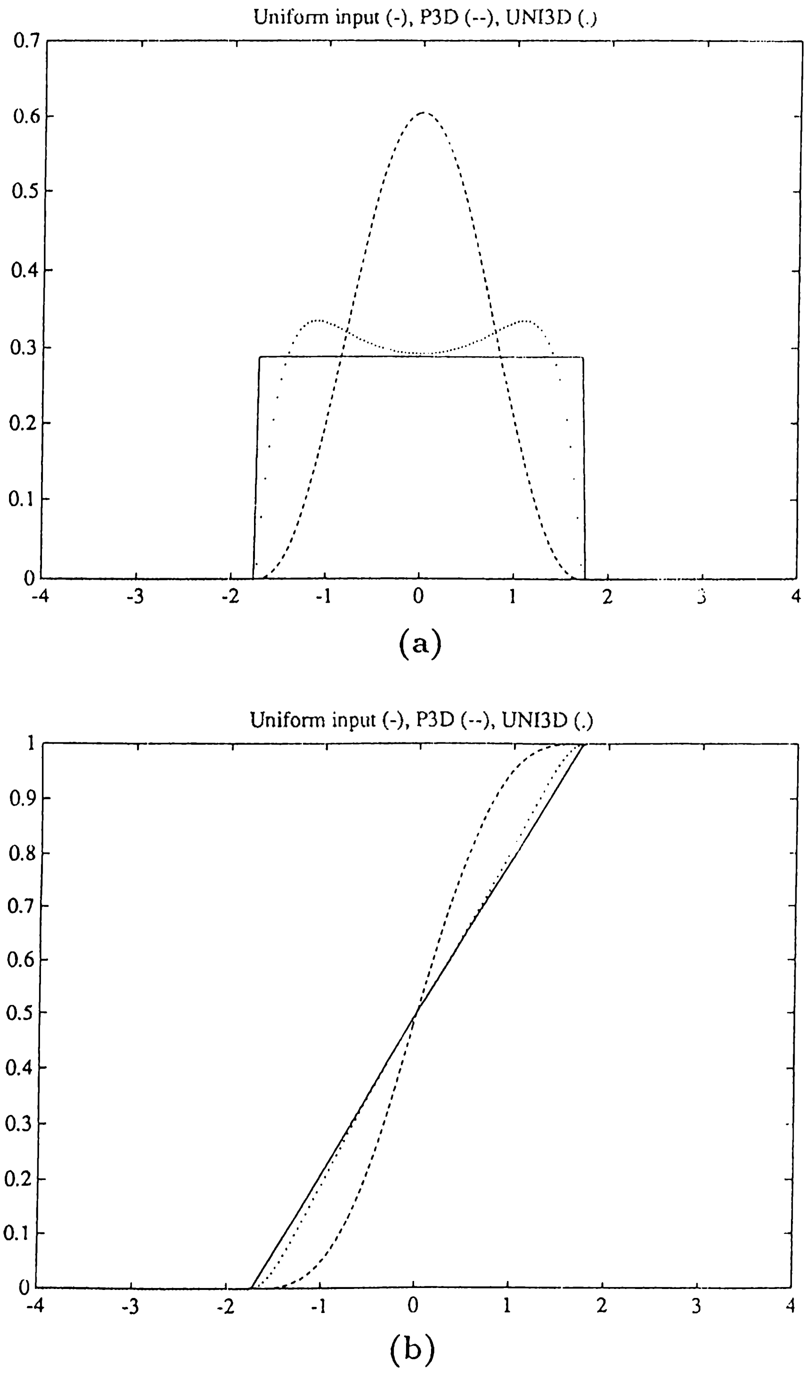
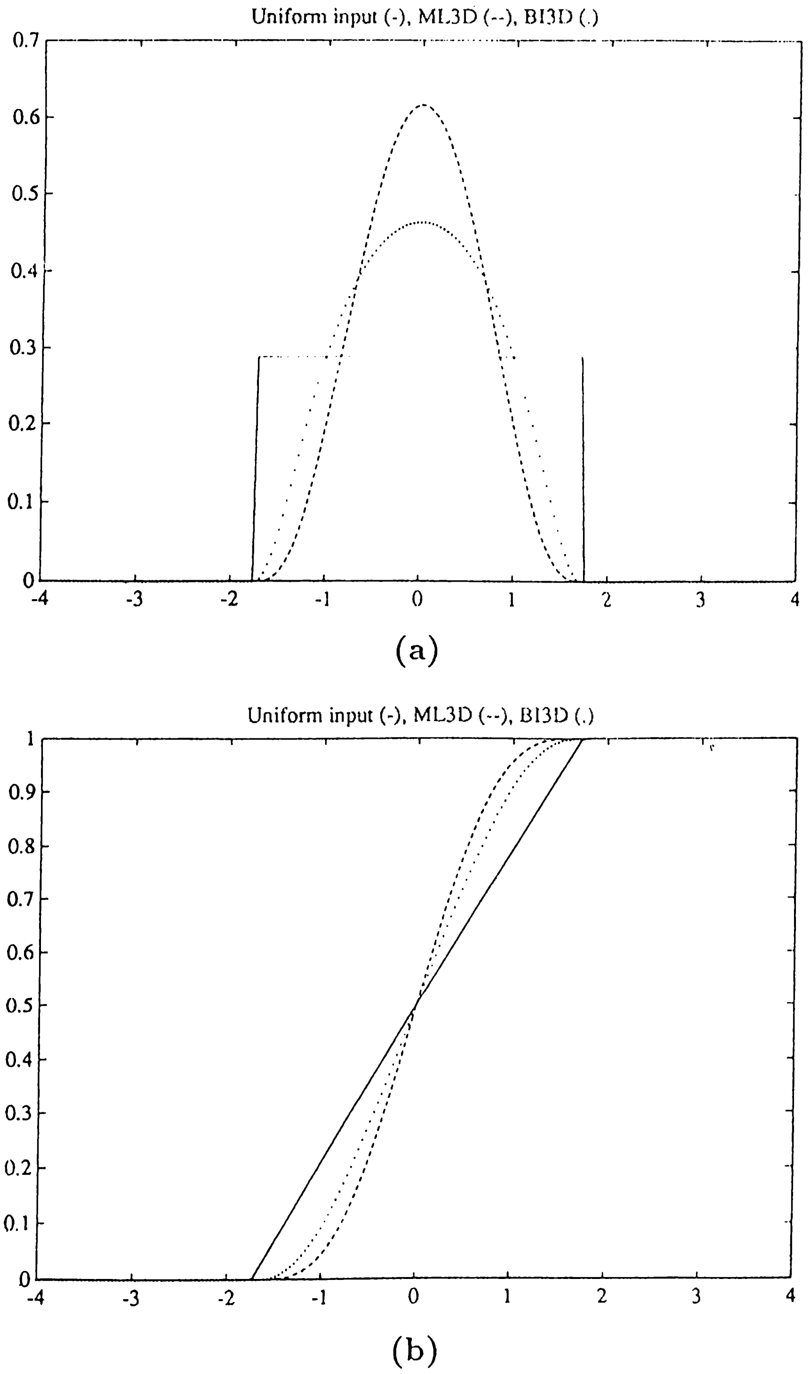


Figure 3.10 The output statistics of P3D, and UNI3D for zero mean, unit variance uniform noise, (a) the probability density functions, (b) the probability distribution functions.



(a)

(b)

Figure 3.11 The output statistics of BI3D, and ML3D for zero mean, unit variance uniform noise, (a) the probability density functions, (b) the probability distribution functions.

3.6 Simulations

3.6.1 DVSR Video Sequencer

The filters that are developed and analyzed are simulated on a VTE DVSR 100 [26] image sequencer (Figure 3.12). The sequencer makes it possible to test algorithms on real image sequences without actually implementing them in hardware. In the simulations, the filtering structures are implemented in the C programming language. The programs are run on a SUN-3 workstation and the resulting sequences are transferred to the image sequencer for storage and display.

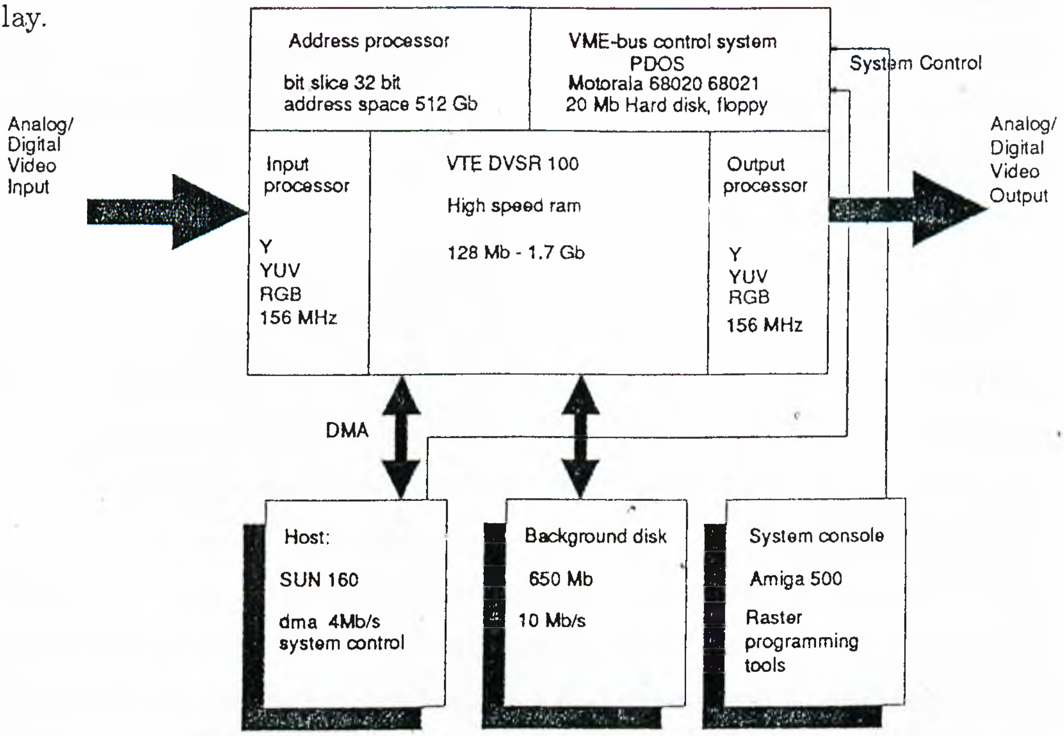


Figure 3.12 The block diagram of the simulation system, VTE DVSR 100.

The sequencer makes it possible to compare results of different algorithms with the original sequence. It is capable of being programmed for different video rasters. The current RAM memory which is 256 Mbytes is expandable to 1.8Gbytes. The sequencer has input and output processors for signal sampling and reconstruction, and a video bus for data transfer. The maximum transfer capability of the video bus is 156Mbits/s. It is possible to process sequences in the Y, YUV, or RGB domains. In image sequence processing, subjective visual quality is as important as the mathematical error measures like the mean square error or the mean absolute error. The visual quality of the filter outputs can easily be evaluated by the display capability of the image sequencer.

3.6.2 Noise Attenuations and Application to Image Sequences

Noise attenuations of all the filters that are defined in Section 3.2 and their recursive versions are calculated for both Gaussian and biexponential independent, identically distributed (i.i.d.) additive white noise using a 4 frame (256 × 128), zero mean, unit variance noise sequence. The results are given in Table 3.1.

The filters are also applied to still and moving image sequences with additive impulsive, Gaussian, and biexponential noise. For impulsive noise, the probability of an impulse is 0.1 with equal probability for positive and negative impulses. For additive Gaussian and biexponential noise distributions, the variance is 30 and the mean is zero. The still image sequence is a 4 frame sequence created using the image "BRIDGE". The motion sequence is a 19 frame sequence called "COSTGIRLS". The mean square error (MSE) and the mean absolute error (MAE) between the original sequence and the filter outputs are given for the "BRIDGE" sequence with impulsive and Gaussian noise distributions in Table 3.2.

The filters are also applied to color image sequences on a scalar basis, i.e., each color component is filtered separately. Parts of the original sequences and the filter outputs are shown in Figures 3.13-3.16 for visual evaluation.

Table 3.1 Output variance of various filters when the input is zero mean, unit variance i.i.d. noise with Gaussian and biexponential distributions.

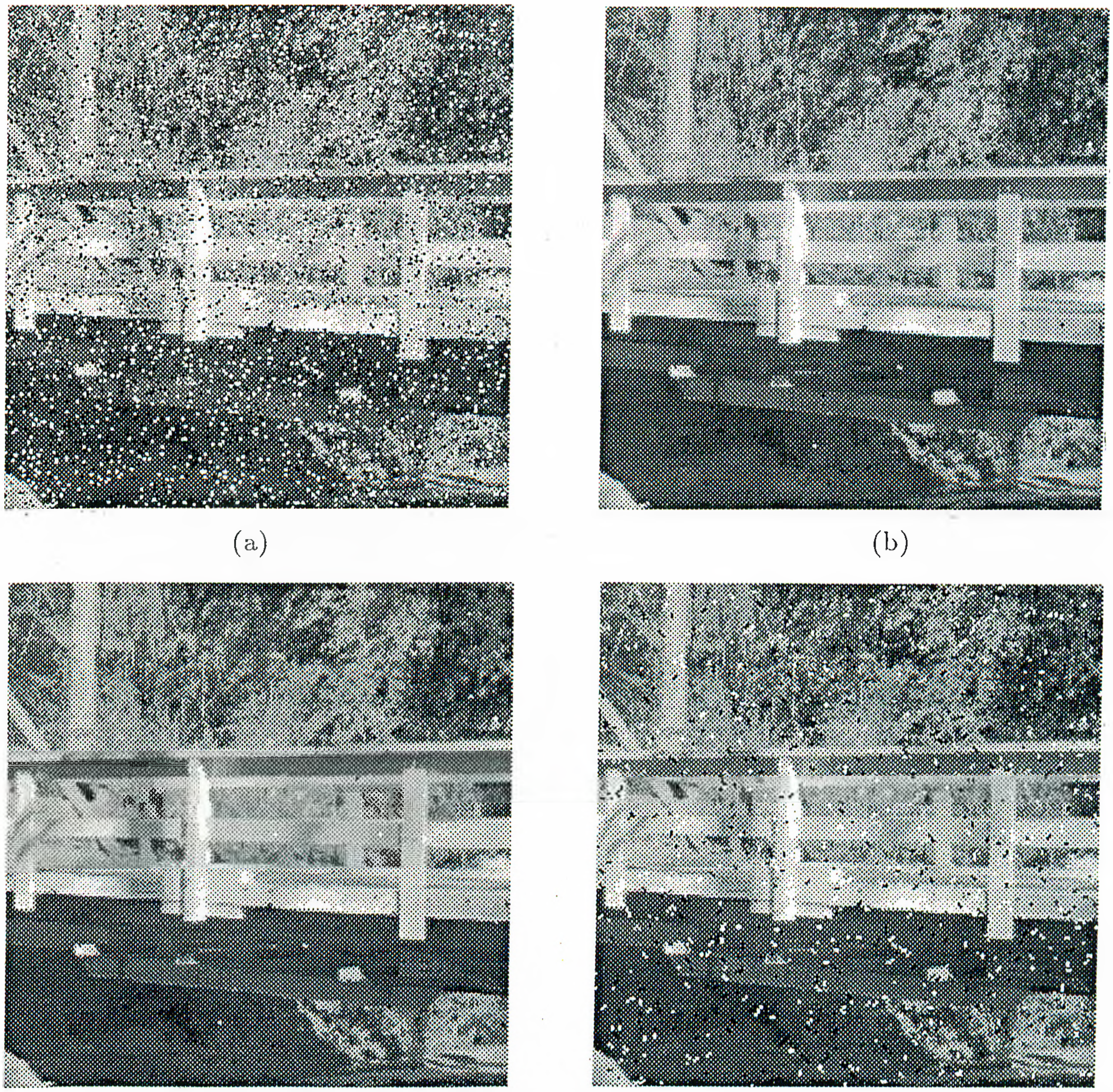
FILTER-TYPE	GAUSSIAN	BIEXPONENTIAL
P3D	0.238	0.137
P3DR	0.117	0.061
ML3D	0.222	0.124
ML3DR	0.119	0.059
PL3D	0.214	0.119
PL3DR	0.081	0.039
CR3D	0.213	0.118
CR3DR	0.080	0.036
UNI3D	0.735	0.579
UNI3DR	0.735	0.579
BI3D	0.363	0.231
BI3DR	0.298	0.181
MLW2D	0.687	0.529
MLW2DR	0.678	0.529
PLW2D	0.520	0.376
PLW2DR	0.518	0.375
CRW2D	0.507	0.367
CRW2DR	0.505	0.366
MEDIAN5	0.293	0.178
MEDIAN5R	0.152	0.083
LAVE	0.113	0.113
LAVER	0.101	0.100

As the simulations show, the two proposed 3-dimensional algorithms (P3D and ML3D) have higher noise attenuation than Arce's unidirectional and bidirectional filters. Although the 2-dimensional 5-point median and linear average filters seem to have better noise attenuation, they are not preferable since they also filter the high frequency details in the image causing blurring. This is why they do not give good results when applied to real image sequences as can be seen from the results presented in Table 3.2. The simulations made on moving image sequences show that the 3-dimensional filters presented here do not disturb the motion content of the image. The only disadvantage of 3-dimensional filters compared to their 2-dimensional counterparts is that they require more memory, two frames in our case. However, with current VLSI technology, the

Table 3.2 MSE and MAE between the original "BRIDGE" sequence and the filter outputs for various noise distributions. For impulsive noise, the probability of an impulse is 0.1 and for Gaussian noise, the variance is 30.

FILTER-TYPE	IMPULSIVE	IMPULSIVE	GAUSSIAN	GAUSSIAN
	MAE	MSE	MAE	MSE
P3D	0.989	37.641	12.544	251.794
P3DR	0.954	22.322	10.619	186.492
ML3D	1.242	34.314	12.368	244.233
ML3DR	1.457	29.047	10.872	194.157
PL3D	1.715	36.670	12.156	236.670
PL3DR	1.950	33.177	10.118	171.959
CR3D	2.828	74.471	13.028	277.217
CR3DR	3.159	73.073	11.414	223.496
UNI3D	6.347	833.632	21.045	666.403
UNI3DR	6.347	833.632	21.045	666.403
BI3D	1.301	106.394	14.909	347.280
BI3DR	1.011	58.936	14.028	308.323
MLW2D	6.074	722.766	20.335	629.180
MLW2DR	6.074	722.766	20.335	629.180
PLW2D	4.643	424.764	17.796	497.709
PLW2DR	4.620	424.919	17.814	498.620
CRW2D	5.683	466.005	18.212	524.930
CRW2DR	5.605	464.819	18.188	523.455
MEDIAN5	5.720	154.661	14.881	359.088
MEDIAN5R	5.421	119.197	12.992	283.006
LAVE	15.184	422.844	12.440	270.612
LAVER	14.589	400.602	12.224	259.358

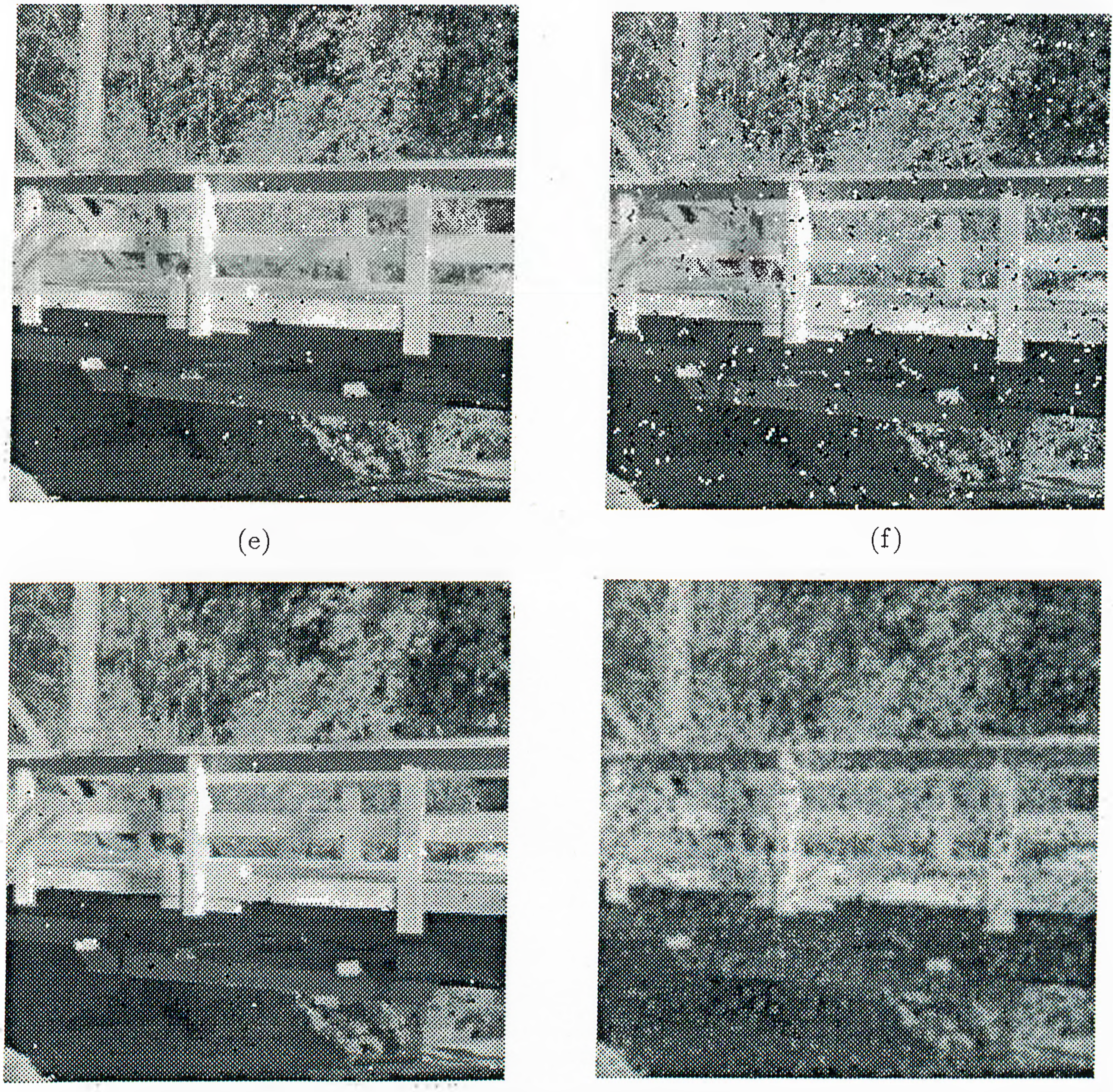
algorithms can already be implemented at video rates [27].



(a) (b)

(c) (d)

Figure 3.13 Part of the original noisy "BRIDGE" sequence and the filter outputs for impulsive noise with probability 0.1. (a) Original image, (b) P3D output, (c) ML3D output, (d) UNI3D output, (e) BI3D output, (f) MLW2D output, (g) MEDIAN5 output, (h) LAVE output.



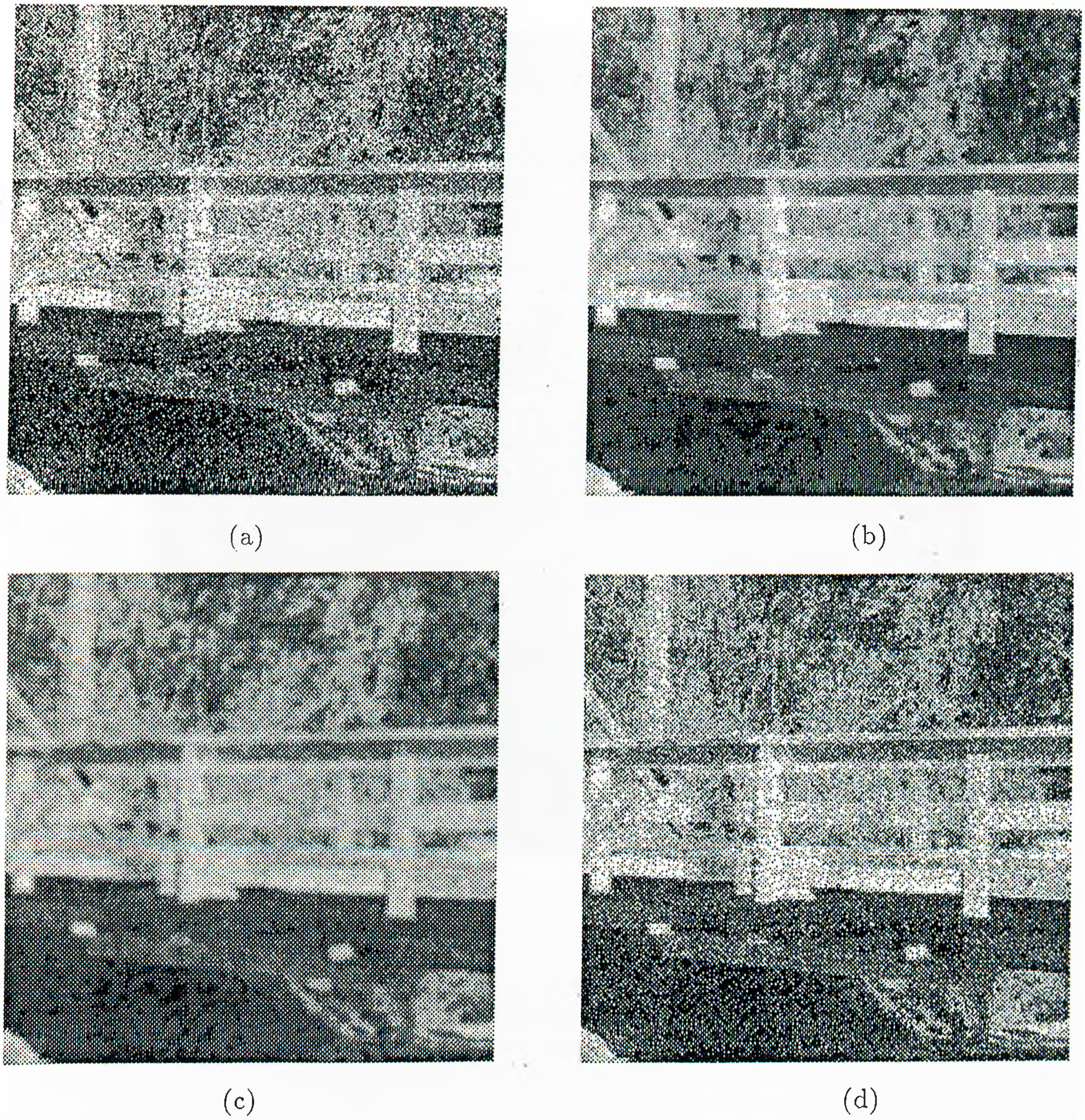
(e)

(f)

(g)

(h)

Figure 3.13 Continued.



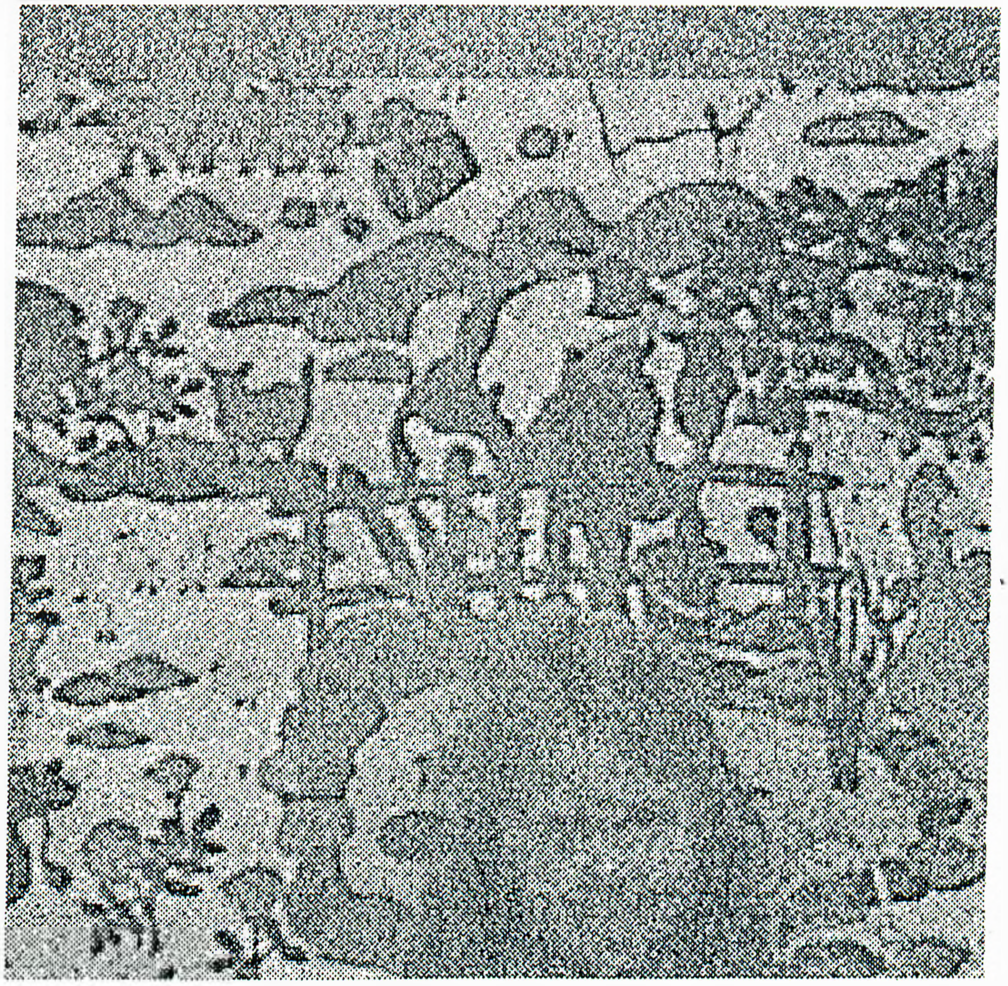
(a)

(b)

(c)

(d)

Figure 3.14 Additive Gaussian noise of variance 30. Parts of (a) the original noisy "BRIDGE" sequence, (b) P3DREC (recursive) output, (c) LAVE output, (d) UNI3DREC (recursive) output.

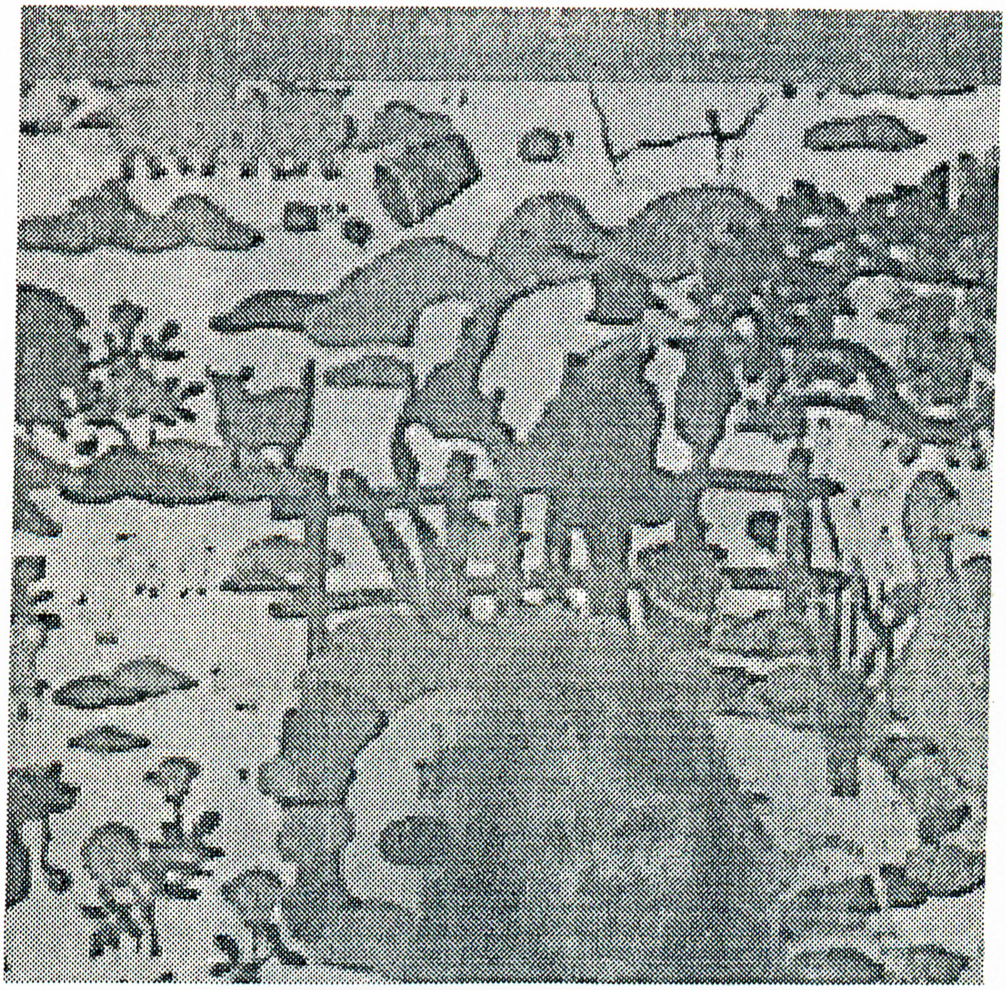


(a)

Figure 3.15 Additive Gaussian noise of variance 30. Part of (a) the original noisy "COSTGIRLS" sequence, (b) LAVE output, (c) P3DREC (recursive) output.

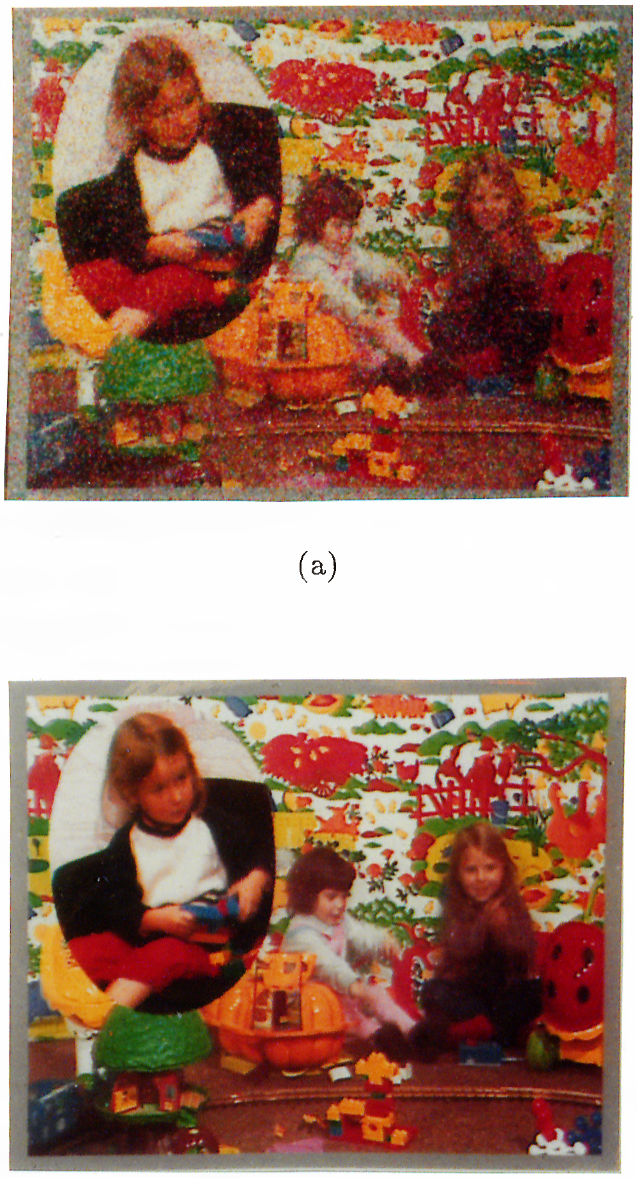
(b)

Figure 3.15 Continued.



(c)

Figure 3.15 Continued.



(a)

(b)

Figure 3.16 Impulsive noise of probability 0.1. Frame 8 of (a) the original noisy "COSTGIRLS" sequence, (b) P3D output.

Chapter 4

IMAGE SEQUENCE CODING

Images and image sequences contain large amounts of data. Because of the limitations in bandwidth and memory, compression is needed for their storage and transmission. Many algorithms have been developed for the compression of images using predictive, interpolative or transform coding methods [28]. Recently advances in TV technology and the introduction of High Definition Television (HDTV) to the consumer market increased the interest in image coding. Active research is still going on in this field.

4.1 General

For high resolution TV applications, transform coding methods have been found to be complex. For this reason, a significant part of the research on TV applications of image coding is focused on predictive and interpolative coding techniques. One main difference between image coding and image sequence coding is the motion content of image sequences. It has been found that a compromise between resolution in time and resolution in space has to be made to obtain good visual quality. The eye is insensitive to high frequency details around moving parts of the image sequence where high resolution in time is critical. In still parts of the image, high resolution in space is essential for acceptable image quality whereas time resolution can be lower. This observation leads to adaptive image coding techniques which the current HDTV image coding systems are based on.

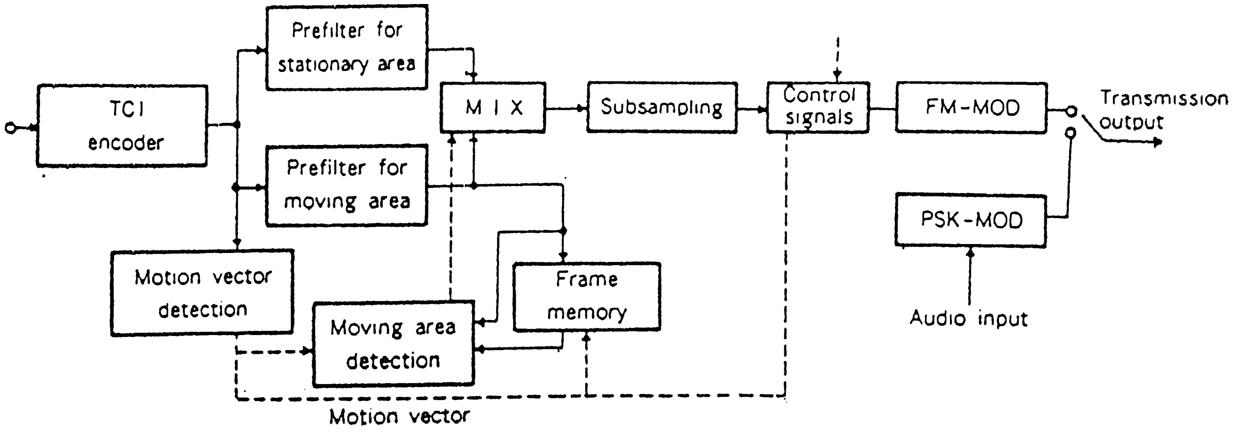
There are basically two main image coding structures developed for HDTV applications. The first one is the Japanese HDTV called MUSE [29]. This system uses a motion detector and two different coding-decoding methods for moving and non-moving parts of the image sequence. The block diagram of the MUSE encoder is shown in Figure 4.1.

The second HDTV system is developed by EUREKA as the European HDTV and is called HDMAC [31]. This system processes the image sequence in three different branches corresponding to no-motion, slow-motion, and fast-motion areas of the image. A different coding-decoding method is used in each branch. The block diagram is shown in Figure 4.2.

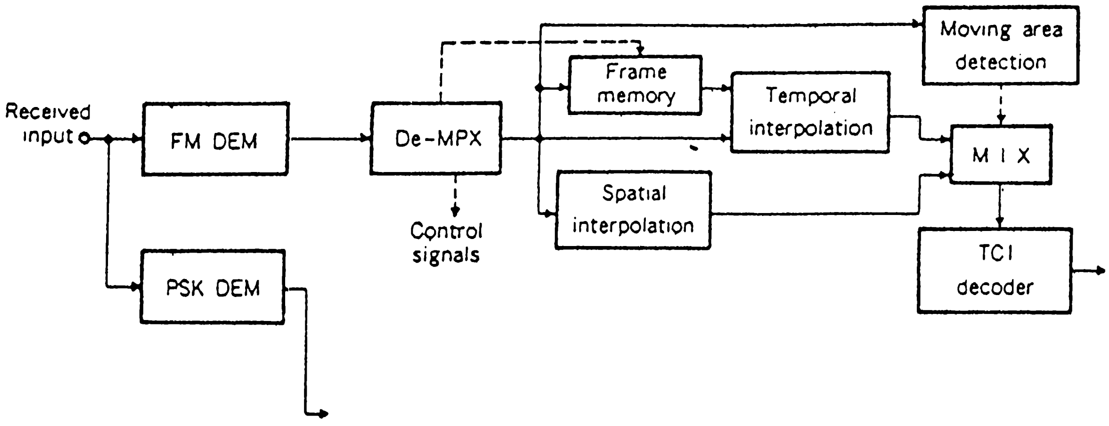
In adaptive image coding algorithms, the performance also depends on the reliability of the motion detector. To have robust systems against motion detector errors, the algorithms developed for each branch should be able to perform relatively well under different conditions. In this thesis, a 3-dimensional median-based interpolative coding-decoding method is developed for no-motion parts of the image which gives acceptable image quality even under slow-motion.

4.2 Median Operation in Image Coding

The detail preserving property of the median operation has been used in image coding as in other areas of image processing. One of the applications of the median operation in image coding is the differential pulse code modulator (DPCM) that uses a median-based nonlinear predictor [32]. The predictor in DPCM systems predicts the new value from the previous ones and only the difference signal between input and the predicted signal is transmitted or stored [28]. In the predictor developed in [32], the median is taken over several subpredictors each of which is optimized for a different ramp or edge signal in varying



(a) Encoder

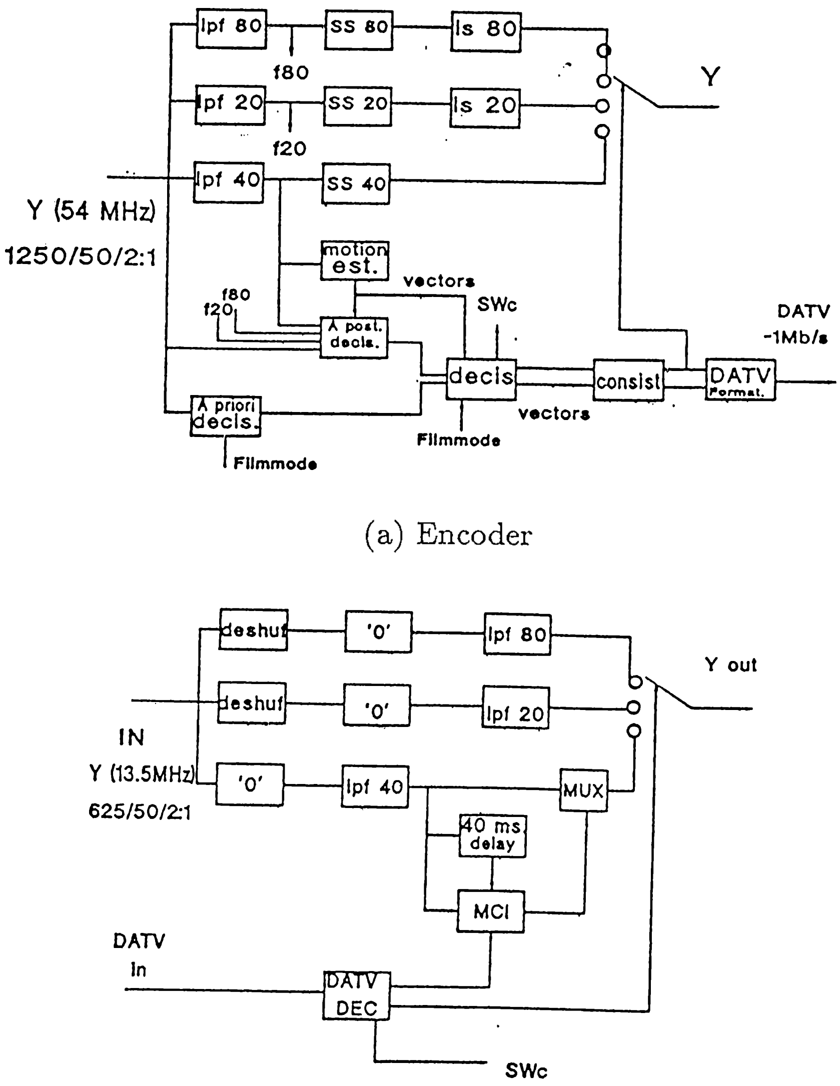


(b) Decoder

Figure 4.1 The block diagram of the MUSE coding-decoding system [30].

orientations. It has been shown that the median operation brings an improvement in prediction errors. Median operation has also been applied to adaptive predictive coding as in [4].

Another image coding method that is commonly used in TV applications is the interpolative coding where only a subset of the pixels are transmitted or stored.



(a) Encoder

(b) Decoder

Figure 4.2 The block diagram of HDMAC coding-decoding system [31].

The missing pixels are interpolated at the receiver using the available ones. This interpolation operation can be linear or nonlinear. Quincunx downsampling is an interpolative coding method where only every other pixel in a line taken from offset positions is stored or transmitted. This way the full horizontal and vertical frequencies are preserved and only the diagonal resolution, to which the human eye is insensitive, gets lower [33]. The application of the median operation to the interpolation of quincunx downsampled images has been widely examined in [34]. Various 2- and 3-dimensional median-based interpolation algorithms are applied to different quincunx downsampling methods and the results are compared. However, the algorithms are basically for still image sequences. The 2-dimensional methods are found to be good for moving image sequences but they lower the resolution. The 3-dimensional algorithms distort the motion content to a large extent although they have a much higher resolution in still parts. In this thesis, a 3-dimensional median-based algorithm is introduced that has the perfect reconstruction property in the case of no-motion and a much better result in the case of slow-motion [35].

4.3 3-Dimensional Interpolative Coding and Decoding Algorithm

In interpolative image coding systems, quincunx downsampling is usually preferred since it preserves full horizontal and vertical resolution, and only reduces the diagonal resolution. There are several different quincunx structures that could be used. The quincunx structure that is used here is field-quincunx downsampling with the addition of an offset from frame to frame [33]. This downsampling scheme is illustrated in Figure 4.3. The offset from frame to frame results in a quincunx structure also in x-t and y-t planes. This makes it possible to make use of all three dimensions in the interpolation process, resulting in perfect reconstruction. The offset quincunx downsampling method has not been used much due to the flicker it causes with many 2-dimensional interpolation algorithms. This artifact is eliminated by the structure developed here.

In what follows, two 3-dimensional median-based interpolation algorithms will be presented. The first one is a novel algorithm based on the multilevel median operation [23] and the second one is the algorithm introduced in [34] that is based on the weighted median operation [17].

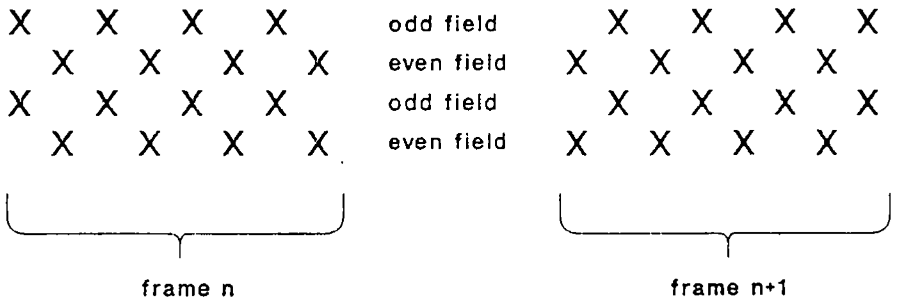


Figure 4.3 3-Dimensional offset quincunx downsampling of the image sequence.

4.3.1 3-Dimensional Multilevel Median-Based Interpolation

Multilevel median operation has been used before to preserve different features in the image sequence. This property of multilevel median operation can be used in the reconstruction process to preserve lines, edges, or ramps in the x-y, x-t, and y-t planes. Based on this idea, the 5-point median operation is applied to each plane of the image sequence separately on the first level of the multilevel structure. The structure used is the same as the 3-dimensional planar filter given in Section 3.2 (Figure 3.1). The reconstructed pixel is taken as the median of the first level outputs. Figure 4.4 shows the 3-dimensional sampling structure. Here, the white pixels correspond to the image points that need to be reconstructed and the dotted pixels correspond to the available image points. Using the notation given in Figure 4.4, the multilevel reconstruction process can be formulated as follows. Let E_1 be the pixel point to be reconstructed. Then the first level filters are given as

$$\begin{aligned} m_{xy} &= MED\,[B_1, D_1, F_1, H_1, (B_1 + D_1 + F_1 + H_1)/4] \; ; \\ m_{xt} &= MED\,[D_1, F_1, E_0, E_2, (E_0 + E_2)/2] \; ; \\ m_{yt} &= MED\,[B_1, H_1, E_0, E_2, (E_0 + E_2)/2] \; , \end{aligned} \qquad (4.1)$$

and the final output is

$$E_1 = MED[m_{xy}, m_{xt}, m_{yt}] \, .$$

Note that for still image sequences, $E_0 = E_1 = E_2$. This gives $m_{xt} = m_{yt}$ which results in perfect reconstruction. Thus the downsampling-upsampling algorithm preserves all high frequency details of still images. When there is slow-motion, the time dimension loses its effect and the term m_{xy} dominates. So the probability of the result being one of the interframe pixels increases resulting in relatively good image quality even in the case of motion.

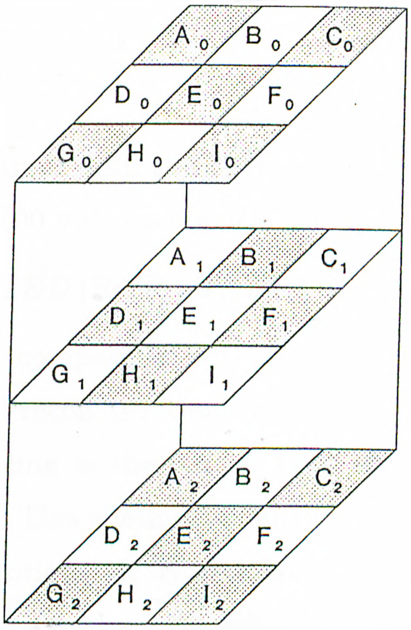


Figure 4.4 3-Dimensional sampling structure corresponding to 3 picture frames.

4.3.2 3-Dimensional Weighted Median-Based Interpolation

This algorithm was first introduced in [34]. The reconstruction method will be given here again for comparison purposes. The algorithm uses the weighted median operation that is defined in Section 2.1. Using the notation in Figure 4.4, the reconstruction filter can be given as follows. Let E_1 be the pixel point to be reconstructed. Then

$$E_1 = MED[B_1, D_1, F_1, H_1, 3 \diamond E_0] ,$$

where $\diamond$ stands for the weighting function. Although the algorithm does not result in perfect reconstruction even in the case of still image sequences, it has a high probability of retaining the correct pixel value since E_0 is given a weight of three, i.e., E_0 is repeated three times in the median operation. Only if E_0 is a minimum or a maximum, it is replaced by the median within the window. This interpolation algorithm does not have the perfect reconstruction property, but it has the advantage of requiring less memory than the 3-dimensional multilevel median-based interpolation.

4.4 Simulations

The downsampling/upsampling algorithms given in Section 4.3 are simulated on the VTE DVSR 100 video sequencer. A 2-dimensional algorithm given as

$$E_1 = MED\,[B_1, F_1, H_1, D_1, (B_1 + F_1 + H_1 + D_1)/4] \tag{4.2}$$

is also simulated to compare it with the other 3-dimensional reconstruction methods. The trivial reconstruction method that could be used with the offset quincunx downsampling is the simple task of retaining the same pixel from the previous frame. This method would also result in perfect reconstruction in the case of no-motion. It is simulated and compared with the other 3-dimensional algorithms to show the difference in the case of slow-motion. Three error measures, mean square error (MSE), mean absolute error (MAE), and the subjective visual quality are used as comparison criteria. The algorithms are applied to the real image sequence called "COSTGIRLS" which is 19 frames long and which has no-motion, slow-motion and fast-motion areas. The motion content of the sequence is obtained using the EUREKA 95-type HDTV motion information. In the error calculations, no-motion corresponds to areas of the sequence moving with speed less than or equal to 0.5 pixels/frame, and slow-motion corresponds to areas of the sequence moving with speeds between 0.5 pixels/frame and 12 pixels/frame. The results for 1/2 compression ratio are given in Table 4.1.

Considering HDTV applications, the same algorithms are applied to obtain 1/4 compression ratios. To obtain this compression ratio, every other frame is skipped during downsampling and reconstructed by simple frame repetition during upsampling. Frame skipping and repetition can only be applied at areas

Table 4.1 Error measures for 1/2 compression ratio.

		multilevel median	weighted median	2-D median	previous pixel repetition
no motion	MAE MSE	1.29 10.53	1.64 21.85	2.23 38.62	1.68 22.30
slow motion	MAE MSE	2.23 39.28	3.15 91.69	2.32 40.32	3.88 132.07

of no-motion and slow-motion since it would cause visible artifacts around fast moving parts. The results for 1/4 compression ratio are given in Table 4.2. The error measures are calculated separately for no-motion and slow-motion areas of the sequence to show the difference between the algorithms with respect to motion.

Table 4.2 Error measures for 1/4 compression ratio.

		multilevel median	weighted median	2-D median	previous pixel repetition
no motion	MAE MSE	2.43 32.38	3.11 74.13	3.10 51.37	3.25 83.95
slow motion	MAE MSE	5.29 161.13	6.53 254.25	5.06 148.13	8.32 388.91

As can be seen in Table 4.1 and Table 4.2, the 3-dimensional multilevel median-based interpolation algorithm gives best results for no-motion areas of the image sequence ($\leq$ 0.5pixels/frame) in both 1/2 and 1/4 compression ratios. Since a large portion of the image sequences that are shown on TV applications have no-motion, this algorithm improves the image resolution to a large extent. Its performance in slow-motion areas is close to the 2-dimensional median based algorithm which is shown to be good for motion [34]. Thus the 3-dimensional multilevel median based algorithm is tolerant to motion detection errors. Moreover, the threshold for motion between no-motion and slow-motion modes of the encoder can be increased if this algorithm is used for no-motion areas in an adaptive encoder. The 3-dimensional weighted median-based interpolation and the previous pixel repetition algorithms show that 3-dimensional algorithms in general distort motion even if they are good for still image sequences. All the algorithms are also applied to color image sequences on a scalar basis, i.e., the algorithms are applied to each color component separately. Parts of the original image sequence and the interpolated outputs are given in Figure 4.5 for visual evaluation.

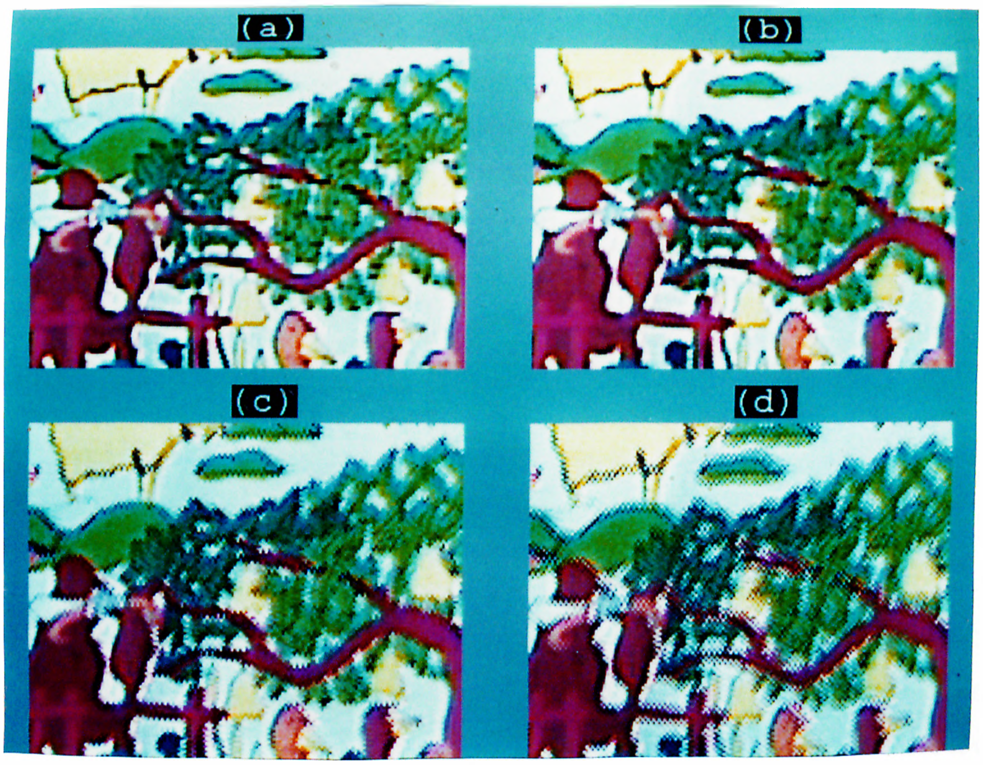


Figure 4.5 Part of frame 8 of (a) the original sequence "COSTGIRLS", (b) the multilevel median-based interpolator output, (c) the weighted median-based interpolator output, (d) the previous pixel repetition algorithm output. The part shown corresponds basically to slow-motion areas of the image sequence.

Chapter 5

CONCLUSIONS

Since the introduction of median filters, median-based algorithms have gained an important place in digital signal processing. They have proved to be a good alternative to linear techniques, especially in image processing tasks. In this thesis the application of the median-based algorithms to two of the main research areas in image sequence processing, namely image sequence filtering and image sequence coding are examined.

Two new 3-dimensional median-based algorithms are introduced for image sequence filtering. They are compared with the other 2- and 3-dimensional algorithms from the literature. The corresponding Boolean functions for the filters are obtained and some observations are made on the root signal structures. The output distributions are derived for the problem of estimating a constant signal under additive white noise. These results are used to plot distributions of the filter outputs under various input noise distributions. In addition to the theoretical analysis, the algorithms developed are evaluated using an image sequencer and real image sequences. Noise attenuations of the filters are obtained and mean square and mean absolute errors are calculated for a real image sequence. From the results we can conclude that median-based algorithms are robust algorithms against the motion in image sequence processing. By the proper choice of the filter structure, the median-based methods give an excellent performance for still and moving images even without a motion detector.

They have an inherent adaptation to the motion content of the image sequence.

In the field of image sequence coding, a 3-dimensional median-based interpolation technique is developed for still and slow moving image sequences. The algorithm developed is compared with several other techniques that could be used with offset quincunx downsampling. The simulations are carried out for 1/2 and 1/4 compression ratios. Mean absolute and mean square errors are calculated for a real image sequence containing different motion areas. It has been shown that the algorithm developed can be integrated to an adaptive coding system or applied on its own.

In both fields of image sequence processing, the algorithms developed are applied to color image sequences on a scalar basis. The application of vector median operation using the same filtering structures may bring an improvement with color processing and should be researched further. Also, directional application of the filtering structures together with a motion detector and estimator could improve the motion performance further. The results obtained show that the nonlinear filtering techniques, of which the median filters are a subclass, offer a large class of choices for image processing which should be considered for future research.

APPENDICES

APPENDIX A

POSITIVE BOOLEAN FUNCTIONS

To simplify the expressions in the derivation of the positive Boolean functions of different filters, the notation given in Figure 3.5 will be used. This notation is explained in more detail in Section 3.3.

Proof of Proposition 2 :

The outputs of the 7-point median filters, PL3D and CR3D, in the first level of 3-dimensional multilevel filter are one if four of the input variables out of seven are equal to one. Thus the PBF's given in the equations (3.25) and (3.26) are obtained by listing all possible combinations of four variables out of seven. The PBF corresponding to ML3D is then computed by taking the 3-point median of the first level outputs and the current pixel value, E_1. Let f_{PL3D}, f_{CR3D} be the first level outputs. Then

$$f_{ML3D} = MED[f_{PL3D}, f_{CR3D}, E_1] = f_{PL3D} f_{CR3D} + f_{PL3D} E_1 + f_{CR3D} E_1 \ ,$$

where

$$\begin{aligned}
f_{PL3D}E_1 = {} & E_0E_1E_2(B_1 + D_1 + F_1 + H_1) + E_1(B_1D_1F_1 + B_1D_1H_1 \\
& + B_1F_1H_1 + D_1F_1H_1) + E_1(E_0 + E_2)(B_1D_1 + B_1F_1 + B_1H_1 \\
& + D_1F_1 + D_1H_1 + F_1H_1) \; ; \\
f_{CR3D}E_1 = {} & E_0E_1E_2(A_1 + C_1 + G_1 + I_1) + E_1(A_1C_1G_1 + A_1C_1I_1 \\
& + A_1G_1I_1 + C_1G_1I_1) + E_1(E_0 + E_2)(A_1C_1 + A_1G_1 + A_1I_1 \\
& + C_1G_1 + C_1I_1 + G_1I_1) \; ; \\
f_{PL3D}f_{CR3D} = {} & A_1B_1C_1D_1F_1G_1H_1I_1 + E_0E_1E_2(A_1 + C_1 + G_1 + I_1)(B_1 + D_1 \\
& + F_1 + H_1) + (E_0E_1 + E_0E_2 + E_1E_2)(A_1C_1 + A_1G_1 + A_1I_1 \\
& + C_1G_1 + C_1I_1 + G_1I_1)(B_1D_1 + B_1F_1 + B_1H_1 + D_1F_1 + D_1H_1 \\
& + F_1H_1) + (E_0 + E_1 + E_2)(A_1C_1G_1 + A_1C_1I_1 + A_1G_1I_1 \\
& + C_1G_1I_1)(B_1D_1F_1 + B_1D_1H_1 + B_1F_1H_1 + D_1F_1H_1) \; .
\end{aligned}$$

Taking the 'or' of the expressions given above results in the equation given in (3.27).

□

Proof of Proposition 3 :

Let z_i, $i = 1 \ldots 5$, be the outputs of the 3-point median operations on the first level of the unidirectional multistage filter. Using equation (3.21), these can be expressed as

$$\begin{aligned}
z_1 = {} & MED[D_1, E_1, F_1] = D_1E_1 \; + \; D_1F_1 \; + \; E_1F_1 \; ; \\
z_2 = {} & MED[A_1, E_1, I_1] = A_1E_1 \; + \; A_1I_1 \; + \; E_1I_1 \; ; \\
z_3 = {} & MED[B_1, E_1, H_1] = B_1E_1 \; + \; B_1H_1 \; + \; E_1H_1 \; ; \\
z_4 = {} & MED[C_1, E_1, G_1] = C_1E_1 \; + \; C_1G_1 \; + \; E_1G_1 \; ; \\
z_5 = {} & MED[E_0, E_1, E_2] = E_0E_1 \; + \; E_0E_2 \; + \; E_1E_2 \; .
\end{aligned}$$

In the binary domain, the maximum operation corresponds to taking the 'or' of the binary input variables and the minimum operation corresponds to taking the 'and' of the binary input variables. Thus z_{max} and z_{min} can be obtained as

$$\begin{aligned} z_{max} =& \max_{1\le i\le 5}[z_i] = z_1 + z_2 + z_3 + z_4 + z_5 \ ; \\ z_{min} =& \min_{1\le i\le 5}[z_i] = z_1 z_2 z_3 z_4 z_5 \ . \end{aligned}$$

The final output is given as the 3-point median of z_{max}, z_{min}, and E_1 as follows.

$$\begin{aligned} f_{UNI3D} =& MED[z_{min}, z_{max}, E_1] \\ =& z_{min}z_{max} \ + \ z_{min}E_1 \ + \ z_{max}E_1 \\ =& z_{min} \ + \ z_{max}E_1 \\ =& A_1B_1C_1D_1F_1G_1H_1I_1E_0E_2 \\ & + \ E_1(A_1 + B_1 + C_1 + D_1 + F_1 + G_1 + H_1 + I_1 + E_0 + E_2) \ . \end{aligned}$$

□

Proof of Proposition 4:

Let z_i, $i = 1 \ldots 4$, be the outputs of the 5-point median operations on the first level of the bidirectional multistage filter. Using equation (3.22), these can be expressed as

$$\begin{aligned} z_1 =& MED[E_0, B_1, E_1, H_1, E_2] \\ =& E_0B_1E_1 \ + \ E_0B_1H_1 \ + \ E_0B_1E_2 \ + \ E_0E_1H_1 \ + \ E_0E_1E_2 \\ & + E_0H_1E_2 \ + \ B_1E_1H_1 \ + \ B_1E_1E_2 \ + \ B_1H_1E_2 \ + \ E_1H_1E_2 \ ; \\ z_2 =& MED[E_0, C_1, E_1, G_1, E_2] \\ =& E_0C_1E_1 \ + \ E_0C_1G_1 \ + \ E_0C_1E_2 \ + \ E_0E_1G_1 \ + \ E_0E_1E_2 \\ & + E_0G_1E_2 \ + \ C_1E_1G_1 \ + \ C_1E_1E_2 \ + \ C_1G_1E_2 \ + \ E_1G_1E_2 \ ; \\ z_3 =& MED[E_0, D_1, E_1, F_1, E_2] \\ =& E_0D_1E_1 \ + \ E_0D_1F_1 \ + \ E_0D_1E_2 \ + \ E_0E_1F_1 \ + \ E_0E_1E_2 \\ & + E_0F_1E_2 \ + \ D_1E_1F_1 \ + \ D_1E_1E_2 \ + \ D_1F_1E_2 \ + \ E_1F_1E_2 \ ; \\ z_4 =& MED[E_0, A_1, E_1, I_1, E_2] \\ =& E_0A_1E_1 \ + \ E_0A_1I_1 \ + \ E_0A_1E_2 \ + \ E_0E_1I_1 \ + \ E_0E_1E_2 \\ & + E_0I_1E_2 \ + \ A_1E_1I_1 \ + \ A_1E_1E_2 \ + \ A_1I_1E_2 \ + \ E_1I_1E_2 \ . \end{aligned}$$

The final output is given as the 3-point median of z_{max}, z_{min}, and E_1 as

$$\begin{aligned} f_{BI3D} =& MED[z_{min}, z_{max}, E_1] \\ =& z_{min}z_{max} + z_{min}E_1 + z_{max}E_1 \\ =& z_{min} + z_{max}E_1 \ , \end{aligned}$$

where

$$\begin{aligned} z_{max}E_1 =& E_0E_1E_2 + E_1(A_1I_1 + B_1H_1 + C_1G_1 + D_1F_1) \\ & + E_1(E_0 + E_2)(A_1 + B_1 + C_1 + D_1 + F_1 + G_1 + H_1 + I_1) \ ; \\ z_{min} =& E_0E_1E_2 + (E_0 + E_1 + E_2)A_1B_1C_1D_1F_1G_1H_1I_1 \\ & + (E_0E_1 + E_0E_2 + E_1E_2)(A_1 + I_1)(B_1 + H_1)(C_1 + G_1)(D_1 + F_1) \ . \end{aligned}$$

By taking the 'or' of the above expressions, we obtain

$$\begin{aligned} f_{BI3D} =& E_0E_1E_2 + E_1(A_1I_1 + B_1H_1 + C_1G_1 + D_1F_1) \\ & + E_1(E_0 + E_2)(A_1 + B_1 + C_1 + D_1 + F_1 + G_1 + H_1 + I_1) \\ & + (E_0 + E_2)A_1B_1C_1D_1F_1G_1H_1I_1 \\ & + E_0E_2(A_1 + I_1)(B_1 + H_1)(C_1 + G_1)(D_1 + F_1) \ , \end{aligned}$$

which is the expression given in (3.29).

□

Proof of Proposition 5 :

The median filters in the first level of the 2-dimensional multilevel weighted median filter, PLW2D and CRW2D, give a weight of three to E_1. Therefore, the outputs of these filters will be one if any one of the binary variables is equal to 1 together with E_1 or all the four variables other than E_1 are equal to 1. Thus the positive Boolean functions corresponding to these filters can be written directly as

$$\begin{aligned} f_{PLW2D} =& MED[B_1, D_1, 3 \diamond E_1, F_1, H_1] \\ =& B_1D_1F_1H_1 + E_1(B_1 + D_1 + F_1 + H_1) \ , \\ f_{CRW2D} =& MED[A_1, C_1, 3 \diamond E_1, G_1, I_1] \\ =& A_1C_1G_1I_1 + E_1(A_1 + C_1 + G_1 + I_1) \ . \end{aligned}$$

The output of the 2-dimensional multilevel weighted median filter is given as the 3-point median of the first level outputs and the current pixel value, E_1.

$$\begin{aligned} f_{MLW2D} =& MED[f_{PLW2D}, E_1, f_{CRW2D}] \\ =& f_{PLW2D}E_1 + f_{CRW2D}E_1 + f_{PLW2D}f_{CRW2D} \\ =& E_1(A_1 + B_1 + C_1 + D_1 + F_1 + G_1 + H_1 + I_1) \\ &+ A_1B_1C_1D_1F_1G_1H_1I_1 , \end{aligned}$$

which is the expression given in (3.32).

□

APPENDIX B

OUTPUT DISTRIBUTIONS

Let $T(\cdot)$ be the threshold function such that

$$T_j\Big(I(x,y,t)\Big) = I^j(x,y,t) = \begin{cases} 1, & I(x,y,t) \geq j \\ 0, & \text{otherwise} \end{cases} ;$$

and

$$T_j\Big(Y(x,y,t)\Big) = Y^j(x,y,t) = \begin{cases} 1, & Y(x,y,t) \geq j \\ 0, & \text{otherwise} \end{cases} ,$$

where $I(x,y,t)$ is the M-valued input sequence and $Y(x,y,t)$ is the M-valued output sequence. For the sake of simplicity in the expressions, the following notation will be used in the proof of the propositions given in Section 3.5.

$$\begin{array}{lll} A_1 = I^{j+1}(x-1,y-1,t) & B_1 = I^{j+1}(x,y-1,t) & C_1 = I^{j+1}(x+1,y-1,t) \\ D_1 = I^{j+1}(x-1,y,t) & E_1 = I^{j+1}(x,y,t) & F_1 = I^{j+1}(x+1,y,t) \\ G_1 = I^{j+1}(x-1,y+1,t) & H_1 = I^{j+1}(x,y+1,t) & I_1 = I^{j+1}(x+1,y+1,t) \\ E_0 = I^{j+1}(x,y,t-1) & E_2 = I^{j+1}(x,y,t+1) & \end{array}$$

Let $F(j)$ be the input probability distribution function, and $Y^{j+1}(x,y,t)$ denote the output of a filter at the $(j+1)$st threshold level. Then, the output distribution function of that filter can be found as

$$F_{out}(j) = Pr\{Y(x,y,t) \leq j\} = Pr\{Y^{j+1}(x,y,t) = 0\} . \qquad (B.1)$$

Using the total probability theorem, this expression can be decomposed into 6 terms as follows. Let

$$\begin{aligned}
P_1 &= Pr\{Y^{j+1}(x,y,t) = 0 | E_0 = E_1 = E_2 = 0\} \ ; \\
P_2 &= Pr\{Y^{j+1}(x,y,t) = 0 | E_0 = E_1 = E_2 = 1\} \ ; \\
P_3 &= Pr\{Y^{j+1}(x,y,t) = 0 | E_0 = E_1 = 0, E_2 = 1\} \\
&= Pr\{Y^{j+1}(x,y,t) = 0 | E_0 = 1, E_1 = E_2 = 0\} \ ; \\
P_4 &= Pr\{Y^{j+1}(x,y,t) = 0 | E_0 = E_1 = 1, E_2 = 0\} \\
&= Pr\{Y^{j+1}(x,y,t) = 0 | E_0 = 0, E_1 = E_2 = 1\} \ ; \\
P_5 &= Pr\{Y^{j+1}(x,y,t) = 0 | E_0 = E_2 = 1, E_1 = 0\} \ ; \\
P_6 &= Pr\{Y^{j+1}(x,y,t) = 0 | E_0 = E_2 = 0, E_1 = 1\} \ .
\end{aligned} \qquad (B.2)$$

Then

$$\begin{aligned}
F_{out}(j) = & P_1 Pr\{E_0 = E_1 = E_2 = 0\} \ + \ P_2 Pr\{E_0 = E_1 = E_2 = 1\} \\
& + P_3 Pr\{E_0 = E_1 = 0, E_2 = 1\} \ + \ P_3 Pr\{E_0 = 1, E_1 = E_2 = 0\} \\
& + P_4 Pr\{E_0 = E_1 = 1, E_2 = 0\} \ + \ P_4 Pr\{E_0 = 0, E_1 = E_2 = 1\} \\
& + P_5 Pr\{E_0 = E_2 = 1, E_1 = 0\} \ + \ P_6 Pr\{E_0 = E_2 = 0, E_1 = 1\}
\end{aligned} \qquad (B.3)$$

The probabilities defined in (B.2) are obtained from the positive Boolean function of the corresponding filter and the results are substituted in (B.3). Finally, the output density function of the filter is obtained from the output distribution function by discrete-time differentiation.

Proof of Proposition 7 :

The output of the +-shaped filter at the first level of 3-dimensional multilevel filter, PL3D, can be expressed at $(j+1)$st threshold level as given in (3.25). Using this equation, the probabilities are obtained as follows.

$$\begin{aligned}
P_1 &= Pr\{B_1 D_1 F_1 H_1 = 0\} \\
&= 1 - Pr\{B_1 = 1 \text{ and } D_1 = 1 \text{ and } F_1 \mp 1 \text{ and } H_1 = 1\} \\
&= 1 - (1 - F(j))^4 \ ; \\
P_2 &= Pr\{B_1 + D_1 + F_1 + H_1 = 0\} \\
&= Pr\{B_1 = 0 \text{ and } D_1 = 0 \text{ and } F_1 = 0 \text{ and } H_1 = 0\} = F(j)^4 \ ;
\end{aligned}$$

$$\begin{aligned}
P_3 &= Pr\{B_1D_1F_1 + B_1D_1H_1 + B_1F_1H_1 + D_1F_1H_1 = 0\}\\
&= Pr\{\text{There are less than 3 ones among } B_1, D_1, F_1, H_1\}\\
&= \binom{4}{0}F(j)^4 + \binom{4}{1}F(j)^3(1-F(j)) + \binom{4}{2}F(j)^2(1-F(j))^2\\
&= 6F(j)^2 - 8F(j)^3 + 3F(j)^4 \ ;\\
P_4 &= Pr\{B_1D_1 + B_1F_1 + B_1H_1 + D_1F_1 + D_1H_1 + F_1H_1 = 0\}\\
&= Pr\{\text{There are less than 2 ones among } B_1, D_1, F_1, H_1\}\\
&= \binom{4}{0}F(j)^4 + \binom{4}{1}F(j)^3(1-F(j))\\
&= 4F(j)^3 - 3F(j)^4 \ ;\\
P_5 &= P_4 \ ;\\
P_6 &= P_3 \ .
\end{aligned}$$

Substituting these expressions in (B.3) results in

$$\begin{aligned}
F_{PL3D}(j) =& F(j)^3\left[1-(1-F(j))^4\right] + (1-F(j))^3\,F(j)^4\\
&+ 3F(j)^2\,(1-F(j))\,(6F(j)^2 - 8F(j)^3 + 3F(j)^4)\\
&+ 3F(j)\,(1-F(j))^2\,(4F(j)^3 - 3F(j)^4)\\
=& F(j)^4\left[35 - 84F(j) + 70F(j)^2 - 20F(j)^3\right] ,
\end{aligned}$$

which is the expression given in (3.45). Since CR3D has the same Boolean function with PL3D when B_1, D_1, F_1, H_1 are replaced by A_1, C_1, G_1, I_1, its output distribution function can be obtained in the same way. For ML3D the probabilities are calculated from the Boolean function given in (3.27) as follows.

$$\begin{aligned}
P_1 &= Pr\{A_1B_1C_1D_1F_1G_1H_1I_1 = 0\} = 1 - Pr\{\text{All of the variables are 1}\}\\
&= \left[1-(1-F(j))^8\right] \ ;\\
P_2 &= Pr\{A_1 + B_1 + C_1 + D_1 + F_1 + G_1 + H_1 + I_1 = 0\}\\
&= Pr\{\text{All of the variables are 0}\} = F(j)^8 \ ;
\end{aligned}$$

$$
\begin{aligned}
P_3 &= Pr\{(A_1C_1G_1 + A_1C_1I_1 + A_1G_1I_1 + C_1G_1I_1)\\
&\qquad (B_1D_1F_1 + B_1D_1H_1 + B_1F_1H_1 + D_1F_1H_1) = 0\}\\
&= Pr\{\text{Either one of the terms in the parenthesis is equal to } 0\}\\
&= 1 - Pr\{\text{Both of the terms in the parenthesis are equal to } 1\}\\
&= 1 - Pr\{B_1D_1F_1 + B_1D_1H_1 + B_1F_1H_1 + D_1F_1H_1 = 1\}^2\\
&= 1 - [1 - Pr\{B_1D_1F_1 + B_1D_1H_1 + B_1F_1H_1 + D_1F_1H_1 = 0\}]^2\\
&= 1 - \left[1 - (6F(j)^2 - 8F(j)^3 + 3F(j)^4)\right]^2\\
&= 12F(j)^2 - 16F(j)^3 - 30F(j)^4 + 96F(j)^5 - 100F(j)^6 + 48F(j)^7 - 9F(j)^8 \ ;
\end{aligned}
$$

$$
\begin{aligned}
P_4 &= Pr\{(A_1C_1 + A_1G_1 + A_1I_1 + C_1G_1 + C_1I_1 + G_1I_1)\\
&\qquad + (B_1D_1 + B_1F_1 + B_1H_1 + D_1F_1 + D_1H_1 + F_1H_1) = 0\}\\
&= Pr\{\text{Both of the terms in the parenthesis are equal to } 0\}\\
&= Pr\{B_1D_1 + B_1F_1 + B_1H_1 + D_1F_1 + D_1H_1 + F_1H_1 = 0\}^2\\
&= \left(4F(j)^3 - 3F(j)^4\right)^2 = F(j)^6\left(16 - 24F(j) + 9F(j)^2\right) \ ;
\end{aligned}
$$

$$
\begin{aligned}
P_5 &= Pr\{(A_1C_1 + A_1G_1 + A_1I_1 + C_1G_1 + C_1I_1 + G_1I_1)\\
&\qquad (B_1D_1 + B_1F_1 + B_1H_1 + D_1F_1 + D_1H_1 + F_1H_1) = 0\}\\
&= Pr\{\text{Either one of the terms in the parenthesis is equal to } 0\}\\
&= 1 - Pr\{\text{Both of the terms in the parenthesis are equal to } 1\}\\
&= 1 - Pr\{B_1D_1 + B_1F_1 + B_1H_1 + D_1F_1 + D_1H_1 + F_1H_1 = 1\}^2\\
&= 1 - [1 - Pr\{B_1D_1 + B_1F_1 + B_1H_1 + D_1F_1 + D_1H_1 + F_1H_1 = 0\}]^2\\
&= 1 - \left[1 - \left(4F(j)^3 - 3F(j)^4\right)\right]^2\\
&= F(j)^3\left(8 - 6F(j) - 16F(j)^3 + 24F(j)^4 - 9F(j)^5\right) \ ;
\end{aligned}
$$

$$
\begin{aligned}
P_6 &= Pr\{(A_1C_1G_1 + A_1C_1I_1 + A_1G_1I_1 + C_1G_1I_1)\\
&\qquad + (B_1D_1F_1 + B_1D_1H_1 + B_1F_1H_1 + D_1F_1H_1) = 0\}\\
&= Pr\{\text{Both of the terms in the parenthesis are equal to } 0\}\\
&= Pr\{B_1D_1F_1 + B_1D_1H_1 + B_1F_1H_1 + D_1F_1H_1 = 0\}^2\\
&= \left(6F(j)^2 - 8F(j)^3 + 3F(j)^4\right)^2\\
&= F(j)^4\left(36 - 96F(j) + 100F(j)^2 - 48F(j)^3 + 9F(j)^4\right) \ .
\end{aligned}
$$

Substituting these expressions in (B.3) results in

$$\begin{aligned} F_{ML3D}(j) =& F(j)^3\left[1-(1-F(j))^8\right] + (1-F(j))^3 F(j)^8 \\ &+ 2F(j)^4(1-F(j))\left(12-16F(j)-30F(j)^2+96F(j)^3\right. \\ &\qquad \left. -100F(j)^4+48F(j)^5-9F(j)^6\right) \\ &+ 2F(j)^7(1-F(j))^2\left(16-24F(j)+9F(j)^2\right) \\ &+ F(j)^4(1-F(j))^2\left(8-6F(j)-16F(j)^3+24F(j)^4-9F(j)^5\right) \\ &+ F(j)^6(1-F(j))\left(36-96F(j)+100F(j)^2\right. \\ &\qquad \left. -48F(j)^3+9F(j)^4\right) \\ =& F(j)^4\left[40-106F(j)+84F(j)^2+60F(j)^3-195F(j)^4\right. \\ &\left. +190F(j)^5-88F(j)^6+16F(j)^7\right] . \end{aligned}$$

□

Proof of Proposition 8 :

The output of the unidirectional multistage filter at the $(j+1)$st threshold level can be expressed as given in (3.28). Using this equation, the probabilities are obtained as follows.

$$\begin{aligned} P_1 &= 1 \ ; \\ P_2 &= 0 \ ; \\ P_3 &= 1 \ ; \\ P_4 &= 0 \ ; \\ P_5 &= Pr\{A_1B_1C_1D_1F_1G_1H_1I_1 = 0\} \\ &= Pr\{\text{At least one of } A_1, B_1, C_1, D_1, F_1, G_1, H_1, I_1 \text{ is equal to } 0\} \\ &= 1 - Pr\{\text{All of them are } 1\} \\ &= 1-(1-F(j))^8 \ ; \\ P_6 &= Pr\{A_1+B_1+C_1+D_1+F_1+G_1+H_1+I_1 = 0\} \\ &= Pr\{\text{All of the variables, } A_1, B_1, C_1, D_1, F_1, G_1, H_1, I_1, \text{ are equal to } 0\} \\ &= F(j)^8 \ . \end{aligned}$$

Substituting these expressions in (B.3) results in

$$\begin{aligned} F_{UNI3D}(j) =& F(j)^3 + 2F(j)^2\,(1 - F(j)) + F(j)\,(1 - F(j))^2 \left(1 - (1 - F(j))^8\right) \\ & + F(j)^{10}\,(1 - F(j)) \\ =& F(j)\left[1 - (1 - F(j))^{10}\right] + (1 - F(j))\,F(j)^{10}\ . \end{aligned}$$

□

Proof of Proposition 9 :

The output of the bidirectional multistage filter at the $(j+1)$st threshold level can be expressed as given in (3.29). Using this equation, the probabilities are obtained as follows.

$$\begin{aligned} P_1 &= 1\ \ ; \\ P_2 &= 0\ \ ; \\ P_3 &= Pr\{A_1B_1C_1D_1F_1G_1H_1I_1 = 0\} \\ &= 1 - Pr\{\text{All of them are } 1\} \\ &= 1 - (1 - F(j))^8\ \ ; \\ P_4 &= Pr\{A_1 + B_1 + C_1 + D_1 + F_1 + G_1 + H_1 + I_1 = 0\} \\ &= F(j)^8\ \ ; \end{aligned}$$

$$\begin{aligned}
P_5 &= Pr\{(B_1 + H_1)(D_1 + F_1)(A_1 + I_1)(C_1 + G_1) = 0\} \\
&= Pr\{\text{At least one of the terms in the parenthesis is equal to } 0\ \} \\
&= 1 - Pr\{\text{All of the terms in the parenthesis are equal to } 1\} \\
&= 1 - Pr\{(B_1 + H_1) = 1\}^4 \\
&= 1 - Pr\{\text{Either } B_1 \text{ or } H_1 \text{ is equal to } 1\ \}^4 \\
&= 1 - (1 - Pr\{\text{Both of them are equal to } 0\})^4 \\
&= 1 - \left(1 - F(j)^2\right)^4 \quad ; \\
P_6 &= Pr\{B_1H_1 + D_1F_1 + A_1I_1 + C_1G_1 = 0\} \\
&= Pr\{\text{All of the pairs that are `and'ed are equal to } 0\} \\
&= Pr\{B_1H_1 = 0\}^4 = Pr\{\text{Either } B_1 \text{ or } H_1 \text{ is equal to } 0\ \}^4 \\
&= (1 - Pr\{\text{Both of them are equal to } 1\})^4 \\
&= \left(1 - (1 - F(j))^2\right)^4
\end{aligned}$$

Substituting these expressions in (B.3) results in

$$\begin{aligned}
F_{BI3D}(j) =& F(j)^3 + 2F(j)^2\,(1 - F(j))\left(1 - (1 - F(j))^8\right) + 2F(j)^9\,(1 - F(j))^2 \\
& + F(j)\,(1 - F(j))^2\left(1 - \left(1 - F(j)^2\right)^4\right) \\
& + F(j)^2\,(1 - F(j))\left(1 - (1 - F(j))^2\right)^4 \\
=& F(j)^3\Big[21 - 80F(j) + 166F(j)^2 - 224F(j)^3 + 202F(j)^4 \\
& 120F(j)^5 + 45F(j)^6 - 11F(j)^7 + 2F(j)^8\Big]\ .
\end{aligned}$$

□

Proof of Proposition 10 :

Using the total probability theorem, the output distribution function of PLW2D can be obtained as follows.

$$
\begin{aligned}
F_{PLW2D}(j) &= Pr\{Y(x,y,t) \le j\} = Pr\{Y^{j+1}(x,y,t) = 0\} \\
&= Pr\{Y^{j+1}(x,y,t) = 0 | E_1 = 0\} Pr\{E_1 = 0\} \\
&\quad + Pr\{Y^{j+1}(x,y,t) = 0 | E_1 = 1\} Pr\{E_1 = 1\} \\
&= Pr\{B_1 D_1 F_1 H_1 = 0\} F(j) \\
&\quad + Pr\{B_1 + D_1 + F_1 + H_1 = 0\} (1 - F(j)) \\
&= F(j) (1 - Pr\{\text{All of the variables, } B_1, D_1, F_1, H_1 \text{ are } 1\}) \\
&\quad + (1 - F(j)) Pr\{\text{All of the variables, } B_1, D_1, F_1, H_1 \text{ are } 0\} \\
&= F(j) \left[1 - (1 - F(j))^4\right] + (1 - F(j)) F(j)^4 .
\end{aligned}
$$

Since CRW2D is the same as PLW2D when B_1, D_1, F_1, H_1 are replaced by A_1, C_1, G_1, I_1, its output distribution function is the same as PLW2D. The output distribution function of MLW2D can be derived in the same way as

$$
\begin{aligned}
F_{MLW2D}(j) &= Pr\{Y(x,y,t) \le j\} = Pr\{Y^{j+1}(x,y,t) = 0\} \\
&= Pr\{Y^{j+1}(x,y,t) = 0 | E_1 = 0\} Pr\{E_1 = 0\} \\
&\quad + Pr\{Y^{j+1}(x,y,t) = 0 | E_1 = 1\} Pr\{E_1 = 1\} \\
&= Pr\{A_1 B_1 C_1 D_1 F_1 G_1 H_1 I_1 = 0\} F(j) \\
&\quad + Pr\{A_1 + B_1 + C_1 + D_1 + F_1 + G_1 + H_1 + I_1 = 0\} (1 - F(j)) \\
&= F(j) (1 - Pr\{\text{All of the variables are } 1\}) \\
&\quad + (1 - F(j)) Pr\{\text{All of the variables are } 0\} \\
&= F(j) \left[1 - (1 - F(j))^8\right] + (1 - F(j)) F(j)^8 .
\end{aligned}
$$

□

REFERENCES

[1] T. S. Huang, Ed., *Image Sequence Analysis*, Springer-Verlag, New York, 1981.

[2] S. S. H. Naqvi, N. C. Gallagher, and E. J. Coyle, "An application of median filters to digital television," ICASSP 86, pp. 2451-2454, Tokyo, 1986.

[3] P. Haavisto, J. Juhola, and Y. Neuvo, " Fractional frame rate up-conversion using weighted median filters," IEEE Transactions on Consumer Electronics, vol. CE-35, no. 4, pp. 272-278, August,1989.

[4] H. Murakami, S. Matsumoto, Y. Hatori, and H. Yamamoto, "15/30 Mbits/s universal digital TV codec using a median adaptive predictive coding method," IEEE Trans. on Communications, vol. COM-35, no. 6, pp. 637-645, June 1987.

[5] G. M. X. Fernando, and D. W. Parker, "Motion compensated display conversion," Proc. 2nd International Workshop on Signal Processing of HDTV, L'Aquila, Italy, February 1988.

[6] K. Öistämö, and Y. Neuvo, "Vector median operation for color image processing," in Proc. of SPIE/SPSE Symposium on Electronic Image and Technology, Santa Clara, USA, Feb. 1990.

[7] T. S. Huang, and Y. P. Hsu, "Image sequence enhancement", in *Image Sequence Analysis*, T. S. Huang, Ed., Springer-Verlag, New York, 1981.

[8] J. W. Tukey, "Nonlinear (nonsuperposable) methods for smoothing data," in Congress Rec. EASCON-74, p. 673, 1974.

[9] G. R. Arce, and E. Malaret, "Motion-preserving ranked-order filters for image sequence processing," Proc. 1989 IEEE Int. Conf. Circuits and Systems, pp. 983-986, Portland, OR, May 1989.

[10] T. A. Nodes, and N. C. Gallagher, Jr. "Median filters: some modifications and their properties," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-30, pp. 739-746, Oct. 1982.

[11] A. Bovik, T. Huang, and D. Munson, "A generalization of median filtering using linear combinations of order statistics," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-31, pp. 1342-1350, Dec. 1983.

[12] Y. H. Lee and S. A. Kassam, "Nonlinear edge preserving filtering techniques for image enhancement," in Proc. 27th Miwest Symp. Circuits Syst., Morgentown, WV, pp. 554-557, June 1984.

[13] P. Heinonen, and Y. Neuvo, "FIR-median hybrid filters," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-35, pp. 832-838, June 1987.

[14] W. K. Pratt, *Digital Image Processing*, New York, NY: John Wiley & Sons, 1978.

[15] N. C. Gallagher, Jr. and G. L. Wise, "A theoretical analysis of the properties of median filters," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-29, pp. 1136-1141, Dec. 1981.

[16] B. I. Justusson, "Median filtering: statistical properties," in *Two-dimensional Digital Signal Processing, II: Transforms and Median Filters*, ch. 4, vol. 42, pp. 161-196, *Topics in Applied Physics*, T. S. Huang, Ed., New York: Springer-Verlag, 1981.

[17] D. R. K. Brownrigg, "The weighted median filter," Commun. ACM, vol. 27, pp. 807-818, Aug. 1984.

[18] J. P. Fitch, E. J. Coyle, and N. C. Gallagher, Jr., "Median filtering by threshold decomposition," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-32, pp. 1183-1188, Dec. 1984.

[19] Multidimensional Signal Processing Committee IEEE Acoust., Speech, and Signal Processing Society, Ed., *Selected Papers in Multidimensional Digital Signal Processing*, The Institude of Electrical and Electronics Engineers, Inc., New York, 1986.

[20] U. Boes, "Linear Filtering in Image Sequences," in *Image Sequence Processing and Dynamic Scene Analysis*, Springer-Verlag, New York,1983.

[21] D. Cano and M. Bénard, "3D Kalman Filtering of Image Sequences," in *Image Sequence Processing and Dynamic Scene Analysis*, Springer-Verlag, New York, 1983.

[22] K. H. Höhne, and M. Böhm, "Processing and Analysis of Radiographic Image Sequences," in *Image Sequence Processing and Dynamic Scene Analysis*, Springer-Verlag, New York, 1983.

[23] A. Nieminen, P. Heinonen, and Y. Neuvo, "A new class of detail preserving filters for image processing," in Proc. 1989 IEEE Int. Conf. Circuits and Systems, Portland, OR, pp. 983-986, May 1989.

[24] O. Yli-Harja, J. Astola, and Y. Neuvo, "Analysis of the properties of median and weighted median filters using threshold logic and stack filter representation," IEEE Trans. Acoust., Speech, Signal Processing, to appear.

[25] E. N. Gilbert, "Lattice Theoretic Properties of Frontal Switching Functions," J. Mathem. Physics, vol. 33, pp. 57-67, 1954.

[26] VTE Digitalvideo GMBH DVSR 100 Documentation.

[27] J. Juhola, P. Haavisto, O. Vainio, T. Raita-aho, and Y. Neuvo, "On VLSI implementation of median based field rate up-conversions," in Proc. 1990 IEEE Int. Symp. on Circuits and Systems, New Orleans, Louisiana, pp. 3042-3045, May 1990.

[28] H. G. Musmann, P. Pirsch, and H.-J. Grallert, "Advances in picture coding," Proc. IEEE, vol. 73, pp. 523-548, Apr. 1985.

[29] Y. Ninomiya, Y. Ohtsuka, Y. Izumi, S. Gohshi, and Y. Iwadate, "Present Status of MUSE", in 2nd International Wórkshop on Signal Processing of HDTV, L'Aquila, Italy, 29 Feb./2 Mar. 1988.

[30] Y. Ninomiya, Y. Ohtsuka, and Y. Izumi, "A single channel HDTV broadcasting system -the MUSE-", NHK Lab. Note, no. 304, 1984.

[31] Ir. F. W. P. Vreeswijk, Dr. M. R. Haghiri, and Dr. C. M. Carey-Smith, "HDMAC coding for compatible broadcasting of high definition television signals," in 16th International TV Symposium, Montreux, 17-22 June 1989.

[32] J. Salo, Y. Neuvo, and V. Hämeenaho, "Improving TV picture quality with linear-median type operations," IEEE Trans. Consumer Electronics, vol. 34, no. 3, pp. 373-379, Aug. 1988.

[33] G. J. Tonge, "The sampling of television images," IBA Experimental and Development Report No. 112/81, May 1981.

[34] T. Jarske, K. Saarinen, J. Juhola, and Y. Neuvo, "Quincunx coding for picture memories," in Proc. 131st SMPTE Tech. Conf., Los Angeles, CA, Oct. 1989.

[35] B. Alp, J. Juhola, Y. Neuvo, and T. Jarske, "Multidimensional reconstruction of quincunx coded image sequences," in Proc. Picture Coding Symposium, Cambridge, MA, pp. 4.5-1 - 4.5-3, Mar. 1990.