6 4057

Non - Linear Filtering Application to Digital Signal Processing

MEDIAN FILTERING: STRUCTURE ANALYSIS AND APPLICATION

A Thesis Submitted by

DEVARAJAN. G

In partial fulfilment of
the requirements for the degree of

DOCTOR OF PHILOSOPHY
OF
THE COCHIN UNIVERSITY OF SCIENCE AND TECHNOLOGY

Department of Electronics
Faculty of Technology
Cochin University of Science And Technology
COCHIN - 682022, INDIA.
1987.

DECLARATION

I hereby declare that the work presented in this thesis is based on the original work done by me under the supervision of Dr. V.K Aatre and Prof. C.S. Sridhar, in the Department of Electronics, Cochin University of Science and Technology, Cochin, and that no part thereof has been presented for any other degree to the best of my knowledge and belief.

Devarajan

(G. DEVARAJAN)

Cochin-682022
January 30, 1987

CERTIFICATE

This is to certify that this thesis is a report of the original work carried out by Mr. G. Devarajan under our supervision and guidance in the Department of Electronics, Cochin University of Science and Technology Cochin-682022, and that no part has been presented for any other degree.

(Dr. V.K. Aatre)
Director
N.P.O.L.
Cochin-682004

(Dr. C.S. Sridhar)
Professor
Department of Electronics
Cochin University of
Science & Technology
Cochin - 682 022

COCHIN
JANUARY 30, 1987.

ACKNOWLEDGEMENT

I profoundly thank Dr.V.K. Aatre Director, Naval Physical and Oceanographic Laboratory Cochin-4, and Prof. C.S. Sridhar Dean, Faculty of Engineering, Cochin University of Science and Technology Cochin-22 for their support, advice, direction and friendship at every stage of this work. They have been a constant source of inspiration to me.

I take this opportunity to thank Prof. K.G. Nair Head, Department of Electronics for extending all the facilities to complete the work in thesis form. Finally my thanks are due to all my friends who helped me to complete this work.

Devarajan

(G. DEVARAJAN)

CONTENTS

ABSTRACT

Median Filtering: Structure, Analysis and Application

Median filtering is a simple digital non-linear signal smoothing operation in which median of the samples in a sliding window replaces the sample at the middle of the window. The resulting filtered sequence tends to follow polynomial trends in the original sample sequence. Median filter preserves signal edges while filtering out impulses. Due to this property, median filtering is finding applications in many areas of image and speech processing. Though median filtering is simple to realise digitally, its properties are not easily analysed with standard analysis techniques.

In this thesis, a new method of characterising Median filters through a matrix operator is introduced. From this a new parameter 'column sum' which gives several features of the signal is extracted The column sum distribution leads to a tree structure from which root paths and state diagrams are evaluated. Theory is developed to get the exact number of passes to reach a root sequence for any given sample sequence.

In addition, two new filters, Fast Convergence Median Filter Interpolation Median Filter are introduced. These two filters are applied in image processing and the results are presented. Theoretical analysis of Median filter on deterministic signals is carried out in frequency domain.

The median filter realisation in terms of minimum hardware for on-line processing is described. In addition, a VLSI

design structure with unit delay is presented. Median filtering is applied in the area of constant false alarm processor, image processing and speech processing and the results are discussed.

In conclusion the median matrix and column sum enable one to extract several interesting properties. FCMF and IMF are very useful in on-line image processing and feature extraction. The ranked operation filter design and its application to underwater target detection is a very useful algorithm.

Chapter - I

INTRODUCTION

One of the objectives in digital signal processing is to design a device or an algorithm to process a sequence of numbers so that the resulting sequence has certain prescribed properties The device or algorithm is called a digital filter. A digital filter can be classified as linear or nonlinear. Linear filter has many properties which simplify the analysis of the same This has allowed a rich theory for design and implementation of linear filters to be developed An operator $\phi(\cdot)$ is said to be linear if $\phi(a\,X + b\,Y) = a.\ \phi(X) + b.\ \phi(Y)$ for any real numbers a and b and inputs X and Y.

Exploiting the superposition property has led to the development of many mathematical tools which simplify the design and analysis of linear filters. For instance a linear system can be represented as the convolution of the input signal with the impulse response of the system. A linear filter representation can be transformed from one domain to another domain that is, time domain to frequency domain or vice-versa. Fourier transform techniques are effective for designing filters when the wanted signal and the unwanted noise are spectrally separate. In short the design techniques for linear filters are well developed and documented.

For some applications, however, linear smoothers are not totally adequate due to the nature of the data being smoothed Added to this the it requires precise definition of filtering' and the object to be filtered. Fig.1.1 shows three examples of data sequences which are to be smoothed Here a linear filter is defined by a sliding window across the input signal. At each position of the window the filter output is determined by some mathematical function operating exclusively on the values in the window In this case averaging of samples in a window is carried out. The examples shown in fig.1.1 exhibit a weakness of a linear smoother for window width 3. This simple averaging smoother shows some of the short-comings of linear filters for the examples illustrated. In the first example where an impulse noise like component is superimposed on the signal, the signal displays sharp discontinuities. Such discontinuities contain much high-frequency energy and are spectrally indistinguishable from noisy component A linear smoother would therefore smear out the sharp changes in the data as well as filter out the noise In Image processing steps are often taken to mark the sharp edges Whenever linear filters are resorted to for such applications it simply changes it into a ramp. Similarly the ramp in the image is blurred. The smeared, blurred output data is not acceptable in many applications. The 'desired' output shown in fig 1 1 is based on what human eye prefers to see in images This forces one to look for a filter other than a linear filter

To meet the 'desired' output shown in fig.1.1 one must contemplate using some type of nonlinear smoothing algorithm which is capable of preserving sharp discontinuities in the data and is still able to filter out noise superimposed on the data. Although such an ideal non-linear smoothing algorithm does not

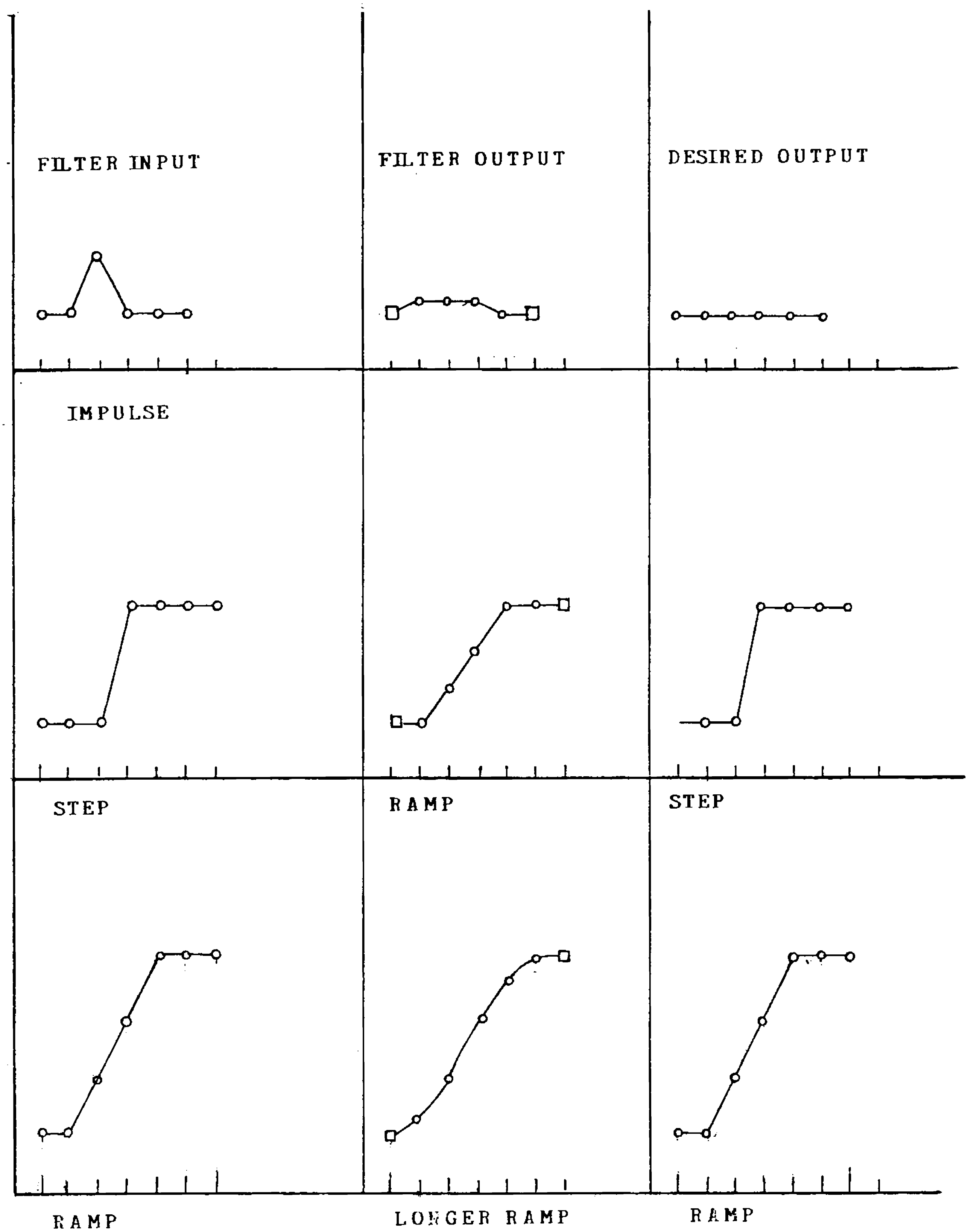


FIG. 1.1. AVERAGING FILTER

exist at present a method proposed by Tukey [1,6] can be shown to have approximately the desired properties. Tukey introduced the median as a robust sliding window filter for smoothing data Despite the known properties of median as an estimator in statistics the mathematics necessary to analyse the effects of median filters on realistic signals are not simple extensions of the existing theory [1,6,9,13,26,27].

Median filters find wide acceptance in the field of image processing and speech processing because of their simple implementation in real time [10,14,21]. As an introduction to the performance of median operator it may be noticed that the "desired" output in fig.1.1 can be derived from a median filter of window size 3 samples The median filter with window size three removes impulses while allowing the edges and ramps to pass unaltered Here the only consideration in median filter selection is the desired window size. In view of these properties median filters have been effectively used in the reduction of high frequency and impulsive noise in digital images [3,7,16,23,24]. Other applications include the smoothing of noise pitch contours in speech signals [2,4] and data compression using root signal properties [5].

The implementation of a median filter requires a simple non-linear operation Let the sampled signal be of length L and a window of width (2K+1) points slide across the signal (K an integer) The filter output at each window position replaces the window center sample by the median sample of the window The start and end effects are accounted for by appending K samples at

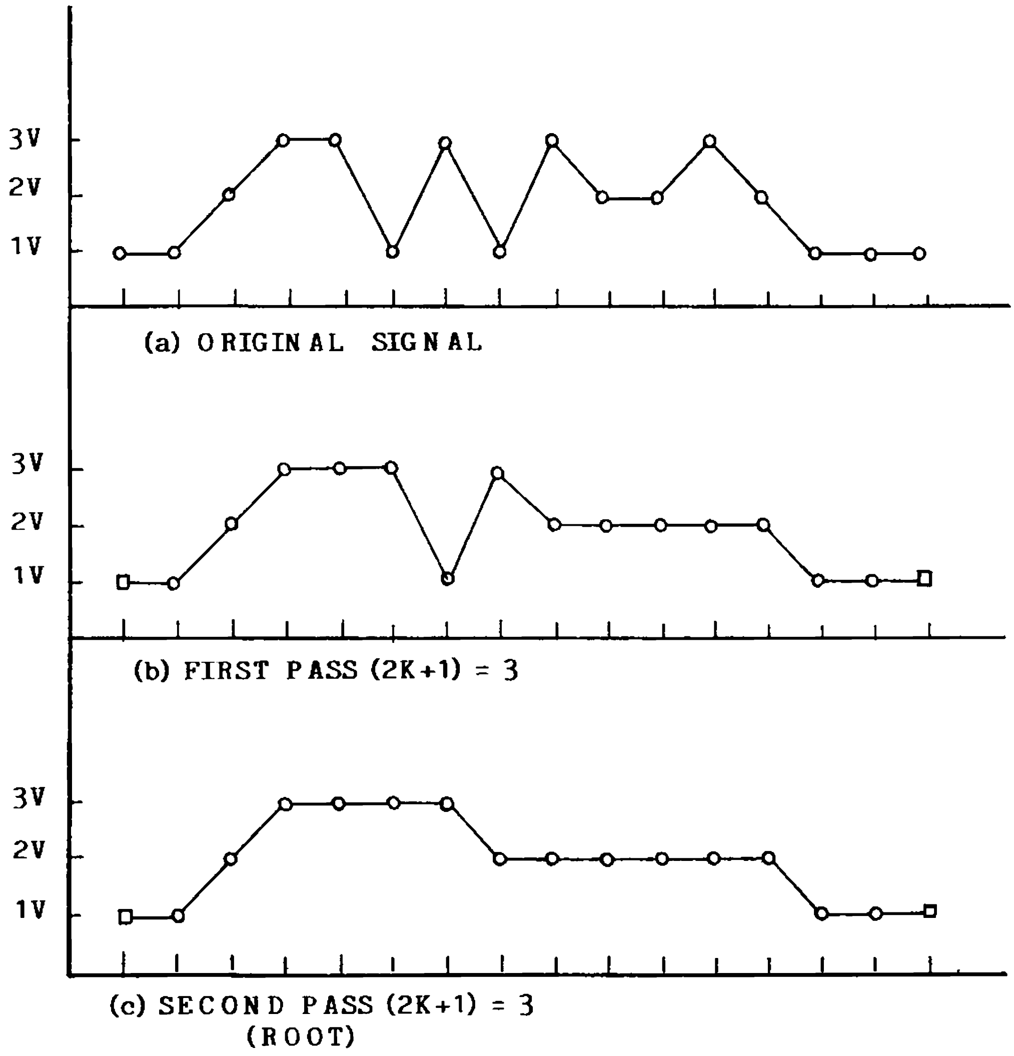


FIG. 1.2. CONVERGENCE OF A MEDIAN FILTER

both the beginning and the end of the sequence as shown in fig 1 2. Thus the basic operation of a median filter is to rank the samples in the window and pick out the middle value as filter output. Median filter is insensitive to spiky noise provided the spike or impulse samples are less than or equal to K for (2K+1) window width It preserves monotonic step edges. These are some of the desired properties in image processing. Median filter finds a place in image processing since it does not blur sharp edges as a linear low pass filter does.

Though median filtering application is increasing for the last decade the necessary mathematical tool for analysis is lacking. Hence the topic 'Median filter structure, Analysis and Applications is relevant to the current area of research. A survey of the work done todate in median filters in analysis and realisation is presented in Chapter II.

This thesis provides methods of characterising median filters and extracting certain properties of the signals. To this end the median operation is expressed in terms of certain matrix transformation and then several properties that depend on the trend (slope change over) of the signal are extracted. Gallagher and Wise [17] have given the bound for the maximum number of passes to reach a root sequence . The number of passes to reach the root sequence is precisely defined and proved These results are discussed and presented in Chapter III.

Chapter IV presents another new area of work in median filtering. An approximation to the running median namely Fast

Convergence Median Filter (FCMF) is described. In order to improve the FCMF results further a method of Interpolation Median Filter (IMF) is developed. FCMF and IMF are compared in performance and implementation with the running median filter. The edge preservation is demonstrated for images with and without noise and compared with (1) Seperable median filter (2) Moving average filter

Frequency domain analysis is carried out in Chapter V. The analysis is restricted to deterministic signals. In all the cases the median filter acts as a SPECTRUM SUBTRACTOR.

Median filtering realisation for on-line processing and certain promising applications are presented in Chapter VI. Median filtering realisation is based on minimum hardware with flexibility to change the window size or to get ranked operation filter without any hardware changes. In addition to this, a possible direct implementation of median filtering technique in VLSI is presented

Finally the thesis is concluded by applying median filtering technique in a Constant False Alarm Rate (CFAR) processor, image processing and speech processing. In CFAR processor running median normalisation is introduced to reduce the variance and is compared with cell averaging algorithm [28, 29, 33]. (In Image processing a relationship between correlation of input and output of median filter is exploited to provide a method of extracting hidden contours The median filtering is applied to extract the speech formant number and formant frequency

Chapter II

PRELIMINARIES

Tukey proposed a non-linear method of signal smoothing exploiting the median property. This non-linearity is different from the nonlinearities of classical electronics. The nonlinear smoother such as running median satisfies

$$\text{Smooth}\ (\lambda\ f(k)) = \lambda\ \text{Smooth}\ (f(k)) \qquad \ldots\ldots \quad (2.1)$$

for all real λ , while conventional nonlinearities often satisfy

$$\text{output}\ (\lambda.f(t)) = \lambda.\text{linear}\ (f(t)) + \lambda^2\ \text{Quadratic}\ (f(t)) + \lambda^3\ \text{Cubic}\ (f(t) + \ldots \qquad \ldots. \quad (2.2)$$

where linear, quadratic, cubic etc are homogeneous of the degree indicated. A conventional nonlinearity satisfying

$$\text{output}\ (\lambda.f(t)) = \lambda.\text{output}\ (f(t) \qquad \ldots\ldots.. \quad (2.3)$$

has only odd terms in equation (2.2) and cannot satisfy the remainder of equation (2.1) without reducing it to the linear term alone. Thus equations (2.1) and (2.2) define very different kinds of limited nonlinearities.

The Median:

The term median has several connotations depending on the context of its use. The median of L numbers, L being odd, of $x(n)$ is the $(\frac{L+1}{2})$ th largest or smallest number ($x(n)$ are all real).

When the numbers $x(n)$ are samples taken from a population, then it is called the Sample Median. David [27], Kendall and Stuart [31] brought out its importance and application in Order Statistics. Another use of median in statistics is the Expected Median. Let $F(x)$ be the c.d.f. of a random variable x. Then the expected median $\hat{Y}$ is the value which satisfies $F(Y)=0.5$. When length of the input sequence $x_{(n)}$ is large compared to window (2K+1) (K an integer) then the output sequence $\{Y_i\}$ of $\{X_i\}$ is obtained such that Y_i is the median of (2K+1) elements of the window centered at X_i. That is

$$Y_i = \text{Median} (x_{(i-K)} \ldots x_i \ldots x_{(i+k)}) \qquad \ldots. (2.4)$$

where $i \geq K$. Such a sequence obtained from x(n) is called Running Median (ϕ). The output $\phi\{X(n)\}$ is always 2K samples shorter due to start and end effects of the window. In order to preserve the length of the sequence, the output sequence is appended with K samples at the start as well as at the end. One way is to append the sequence with K samples at the beginning and at the end with the first and the last samples of the sequence x(n), respectively. Another useful way is to append the median sequence y_i with the first and last median samples. The latter has the advantage of order preserving or order reversing transformation which will be discussed in Chapter III.

2.1 One Dimensional Median Filters

The median of L samples $X_1, X_2 \ldots. X_L$ arranged in ascending or descending order, for L odd, is the central sample. For L even it is the mean of the two central samples. For L even other definitions can be found in literature. In our discussion it is always assumed that the sequence length is odd. The median Y of $\{x_i\}$ is expressed as

$$Y = \phi (X_0, X_1 \ldots \ldots X_L) \text{ where } \phi \text{ is the median operator.}$$
2.2

Example: Let the sample sequence be (3,7,1,Ø,5,2,9). The median is 3, whereas the mean value is 3.8571428. It is clear from this that the median is always a subset of the input sequence whereas the mean is not necessarily so

Definition: The median is the central sample of ranked $\{x_l\}$. If $\{x_l\}$ is arranged in ascending or decending order.

i.e. $x_{l-k} \leq \ldots \leq x_l \leq \ldots \leq x_{l+k}$

then median $(x_{l-k}, \ldots x_l \ldots x_{l+k}) = x_l$ ... (2.5)

In filtering application the median of the sequence replaces the central sample. In applications of median filtering to speech and images, a window (2K+1) is moved along the samples. The window is moved along the sampled values of the signal (or images) from left to right and the median of the samples within the window is computed. The median value computed for this window position replaces its central sample. As the window slides from left to right one new sample enters the window as the oldest comes out of the window and the median is obtained for the entire set of input samples. The median obtained like this is called Running Median or Moving Median. In general the running median can be expressed for (2K+1) window as

$$Y_i = \Phi(X_{(i-k)} \ldots . X_i \ldots .. X_{(i+k)})$$

TWO DIMENSIONAL MEDIAN FILTER:

Digital pictures can be represented by row X column pixels where each pixel is represented by a number equivalent to its grey level.Conventionally a rectangular (or square NXN) window is used in two dimensional median filtering. The intensity at every point in the image is replaced by the median of the intensities of the points contained in the NXN window centered at the point. A two dimensional median filter of NXN window on a picture

$$\{x_{ij}, \; i\,j \in Z^2\} \text{ is defined}$$

$$Y_{ij} = \Phi\{x_{ij}\} = \Phi\left\{x_{i\pm p,\, j\pm q},\; p,\, q \quad \text{window}\right\}_{i,j \,\in\, Z^2} \qquad .... \qquad (2.6)$$

Fig.2.1 illustrates a two dimensional median filtering operation by using a filter window (3X3) moving from left to right till the processing matrix Z covers all the pixels. The two dimensional window is moved from left to right and the median sample replaces the window central sample at every position. When the window reaches extreme right, it is brought back to the beginning and pushed down by one row. This process is repeated for the entire pixel matrix. The start and end delay of the window is compensated by (1) appending the first and the last median samples or (2) the input samples as such for the delay or (3) a smaller window in the border.

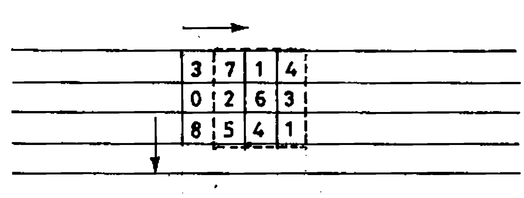


Fig. 2.1 Filter Window (3 x 3)

Different shapes of windows can be used viz. line segment, cross, square, tilted square, circle etc. A window is so chosen that number of elements within it is always odd and symmetry is

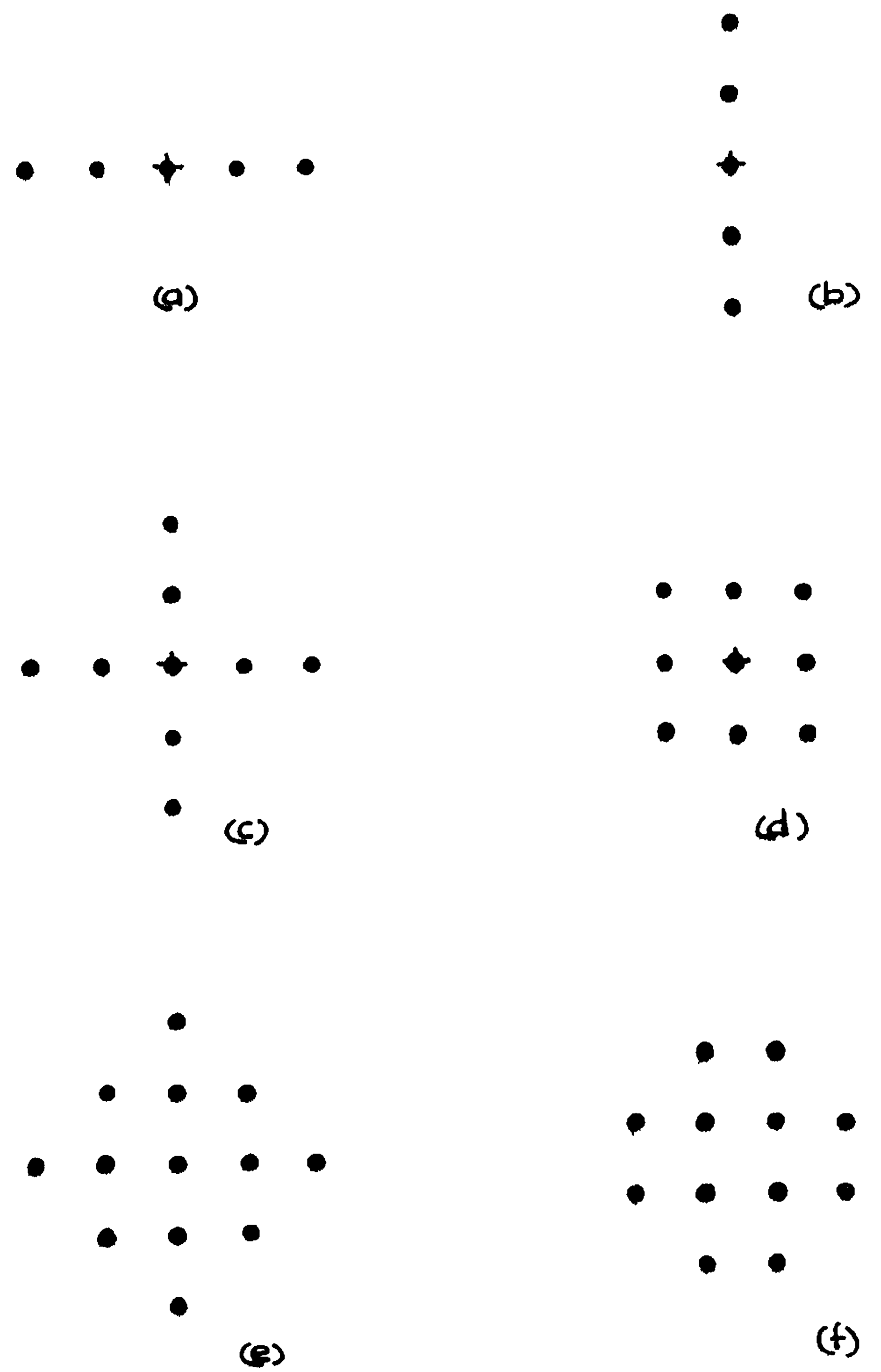


Fig.2.2 MEDIAN WINDOWS (a),(b) LINE SEGMENTS
(c)CROSS (d)SQUARE (e)TILTED SQUARE
(f)CIRCLE AND CIRCLE RINGS.

maintained in both the axes with respect to its center. Some of the window shapes are shown in fig. 2.2.

All these windows have wide application [16,18] in image processing. The line segments shown in fig. 2.2(a) and (b) are useful for one dimensional processing. Huang et.al [11] and Narendra [16] have applied a variant of the median filter - the separable median filter, for image noise smoothing. The separable median filter is a two dimensional non linear filter derived from successive applications of one dimensional median filter of size (2K+1), applied first along the rows and then along the columns of an image (or vice-versa). The windows (2K+1) applied for separable median filters are one dimensional windows. The major advantages of the separable formulation are : faster computer realisation and simple real time hardware implementation.

2.2 Properties of running median.

Running median has several good properties which makes it a strong candidate for a smoother. Rabiner et. al. [2] and Tyan [8,18] have pointed out some of its deterministic properties.

(i) Scaling

$$\text{Median}\{\alpha x_{(n)}\} = \alpha\,\text{Median}\{x_{(n)}\} \quad \ldots \quad (2.7)$$

where α is a real constant. Further

$$\text{Median}\{\alpha + x_{(n)}\} = \alpha + \text{Median}\{x_{(n)}\} \quad \ldots \quad (2.8)$$

(ii) Median Filtering is time invariant

(iii) Median filtering does not smear out sharp discontinuity as long as the duration of a discontinuity does not exceed a critical value. This is not true for linear filtering.

(iv) Median filtering in general does not obey superposition property i.e.

$$\text{Median } \{ a.\, x_{1(n)} + b.\, x_{2(n)} \} \neq a.\, \text{Median } \{x_{1(n)}\} + b.\, \text{Median } \{x_{2(n)}\} \quad \ldots\ldots \quad (2.9)$$

where a and b are constants and $x_{1(n)}$, $x_{2(n)}$ are two input sequences.

Tyan [18] has brought out some interesting deterministic properties from equation (2.4).

Property 1: If $X_{-K} \leq \ldots \leq X_{\emptyset} \leq \ldots \leq X_{K}$

then median $(X_{-K} \ldots\ldots X_{\emptyset} \ldots\ldots\ldots X_{K}) = X_{\emptyset}$

Property 2 : If g(x) is monotonic, then

median $(g(x_1) \ldots\ldots g(x_{2K+1})) = g\ [\text{median } (x_1 \ldots\ldots x_{2K+1})$

With the definition of median from equation (2.4) and from property 1 it can be seen that a 'monotonic sequence', i.e. a sequence such that $Xn \leq Xm$ for all $n \leq m$ is "invariant" to a median filter of arbitrary window length. Monotonic sequences/low order polynomials are the simplest invariant (fixed) points of median filters and generalisations of these constitute more important class of invariant points (roots).

Scaling the signal does not affect median filters' performance. This property will be more useful in processing two dimensional data.

As mentioned in the preceeding paragraph, monotonic sequences are fixed points of median filters of arbitrary window length; however the requirement of monotonicity is unnecessarily restrictive. Since the median filter is of fixed window length, the monotonicity can be relaxed. Tyan was the first to point out their importance and to deduce many important theorems about their properties under median filtering.

A sequence {xn} is locally monotonic of length m (abbr. LOMO (m)) if and only if {Xn, Xn+1 ... Xn+m-1} is monotonic for a given n.

A LOMO (m) sequence is also LOMO(p) provided $p \leq m$. If there is any change in the trend, then a LOMO(m) sequence must stay constant for atleast m-1 samples.

The following two theorems reveal the importance of locally monotonic functions:

Theorem 2.1: A LOMO(m) sequence is invariant to running median filter of window (2k+1) for all K provided $K \leq m-2$. This implies $\Phi_{(2K+1)} \{X_n\} = \{X_n\}$.

Theorem 2.2: If $\{X_n\}$ is a fixed point of $\Phi_{(2K+1)}$ and if there is monotonic segment $\{X_p, X_{p+1} \ldots X_{p+k})$ of length (K+1), then $\{X_n\}$ is LOMO (K+2).

The preceeding two theorems state that if a fixed point 2K+1 is smooth enough for a segment of length (K+1), then it is smooth over the whole length (i.e. LOMO (K+2)).

Theorem 2.3: If $\{X_n\}$ is a fixed point (subject to appended values at the edges same as input signal of ϕ_{2K+1}) and if it is nowhere LOMO (K+1), then $\{X_n\}$ is a bivalued sequence i.e. $\{X_n\}$ can take on only two values alternatively.

Tyan has classified the fixed points into two groups. The fixed points defined in theorem 2.2 and theorem 2.3 are called Type I and Type II fixed points respectively. Gallagher and Wise [17] have defined input sequence structures. These signal sequence structure definitions are used for median filtering structure and analysis in Chapter III. For a finite sample $\{X_n\}$ of length L quantised to q levels, different signal structures definitions are:

(1) A CONSTANT NEIGHBORHOOD is a region of at least (K+1) consecutive points all of which are identically valued points.

(ii) An EDGE is a monotonic region between two constant neighborhoods of different values. The connecting monotonic region cannot contain any constant neighborhood.

(iii) An IMPULSE is a set of K or less points whose values are different from the surrounding regions which are identically valued constant neighborhoods.

(iv) An OSCILLATION is a sequence of points which is not part of a constant neighborhood, an edge or an impulse.

(v) A Root is an 'invariant' signal which is not modified by median filtering (Fixed points).

To illustrate the preceeding definitions, an example is shown in fig.2.3. Let the window be 3 (i.e. K=1). In this sequence, at length 4 an edge which is separated by two

4 4 4 3 1 1 6 3 3 3	3 5 2 5 2 5 2	Input sequence
□ 4 4 3 1 1 3 3 3 3	3 3 5 2 5 2 □	MF output

Fig. 2.3

neighborhoods of different values is present. At length 7 an impulse is present. Beyond length 11, the input sequence contains only oscillations. The start and end of the output sequence is marked as □ which are the appended samples. The window is sliding over the input sequence and the median sample for the window replaces the central sample of the window. The output sequence obtained in this manner will have 2K samples less than the input sequence. The reason for this is due to the start and end effects of a window on the input sequence. The output sample do not alter upto the length 7 The reason for this is that MF will not alter a neighborhood or an edge of an input sequence. At the input sequence length 7, there is an impulse. This impulse is wiped out in the output. The trend of the input sequence following sample 12 is changing alternatively. This signal is oscillatory. This portion of the signal is the one which undergoes changes in median filtering. In such cases also it is possible to get an invariant output (Root) by repeated passage through a median filter. This can be seen in the following example for window 3(K=1).

4	4	4	3	1	1	6	3	3	3	3	5	2	5	2	5	2	INPUT SEQUENCE
								(a)									
$\boxed{4}$	4	4	3	1	1	3	3	3	3	3	3	5	2	5	2	$\boxed{2}$	FIRST PASS
								(b)									
$\boxed{4}$	4	4	3	1	1	3	3	3	3	3	3	3	5	2	2	$\boxed{5}$	SECOND PASS
								(c)									
$\boxed{4}$	4	4	3	1	1	3	3	3	3	3	3	3	3	2	2	$\boxed{2}$	THIRD PASS
								(d)									

Fig. 2.4

It can be observed in fig. 2.4(b) to 2.4(d) that the oscillatory portion of the signal undergoes changes for each pass. The beginning and the end values of the oscillation/trend change over point is reduced by forming neighborhoods on both the side of the oscillations. Thus after the third pass, the oscillation ceases and becomes neighborhoods only. This signal is invariant to further median filtering and is called a 'ROOT'.

In Fig.2.4(a) the start and end samples of oscillation viz. 13 and 18 are 5 and 2, respectively. This part of the signal has become two neighborhoods after three passes. However, if the start and the end sample values are same it is possible to get only one neighborhood for the entire length of the oscillatory sequence. The root signal sequence is not unique. It is a function of window size. Consider the input sequence given in fig.2.4(a) for window size 5 (K=2). In fig.2.5 it is shown that two successive

4	4	4	3	1	1	6	3	3	3	3	5	2	5	2	5	2	INPUT SEQUENCE
$\boxed{4}$	$\boxed{4}$	4	3	3	3	3	3	3	3	3	3	3	5	2	$\boxed{2}$	$\boxed{2}$	FIRST PASS
$\boxed{4}$	$\boxed{4}$	4	3	3	3	3	3	3	3	3	3	3	3	2	$\boxed{2}$	$\boxed{2}$	SECOND PASS

Fig. 2.5 MF Output for window size 5

passes of the signal produces a root sequence. Here it is to be noted that the neighborhood portion of the signal sequence for a window 3 (K=1) is oscillatory for window 5 (K=2). In the first pass the number of trend changeover points is reduced to 3 from 9. The second pass output is an invariant (Root) signal and it is having extended neighborhoods. The root signal sequence is changing as the window size changes. But the root sequence of larger window MF is always a subset of root sequence of a smaller window MF. The following theorems are presented as a result of the preceeding discussion.

Theorem 2.4 : Given a q level sequence of length L, the necessary and sufficient condition for the signal to be invariant under MF is that the extended signal consist only of neighborhoods.

Theorem 2.5 : Given a q level sequence of $x(n)$, the root sets R_i ,(where R_i is MF output for the ith pass) are nested such that

$$R_i \subseteq R_{i-1} \subseteq \ldots \subseteq R_\emptyset = x_{(n)} \qquad \ldots\ldots (2.1\emptyset)$$

Gallagher and Wise [17] proved that successive median filtering (i.e. the filtered output is itself again filtered through the same filter) eventually reduces the original signal to a root signal. Given a non-root signal of length L, it will become a root after a maximum of 1/2 (L-2) passes for L even and 1/2 (L-1) passes for L odd. In this analysis, it is to be noted that they have only stated the bound for the maximum number of passes to arrive at a root signal. Further work to arrive at the exact number of passes for a root sequence is given in ChapterIII.

Arce and Gallagher [20] developed a theory for the root signal sets of median filters and obtained a tree structure for the root signal set for binary signals. The tree structure for the MF of window size 5 is shown in fig.2.6. The root signal can be built with the first bit as either 0 or 1 and several possible combination of subsequent bits lead to 'allowable root paths'. Such allowable root paths are shown in the tree diagram, the root paths themselves branching into a subtree. From a close look at this tree structure, one can observe that sections of the tree repeat themselves as the tree propagates. This observation leads to the concept of the existence of discrete states. Four different states can be identified and they are:

State A: Two successive digits with values zero (0,0). The next digit can take any value 0 or 1.

State B: Two successive digits where the first has value 0 and the second value of 1 (0,1). The next digit can only be a 1.

State C: Two successive digits where the first has value 1 and the second of 0 (1,0). The next digit can only be a 0.

State D: Two successive digits with value 1 (1,1). The next digit can take any value 0 or 1.

Fig.2.6 shows how these states propagates as the signal lengths increase. Each state will generate other states as can be seen in the state diagram (fig.2.7). The number of roots for a given sequence length l can be written from the state diagram as (Ref. fig.2.7).

$$R(l+1) = 2A(l) + 2D(l) + B(l)+C(l) \qquad \ldots\ldots (2.11)$$

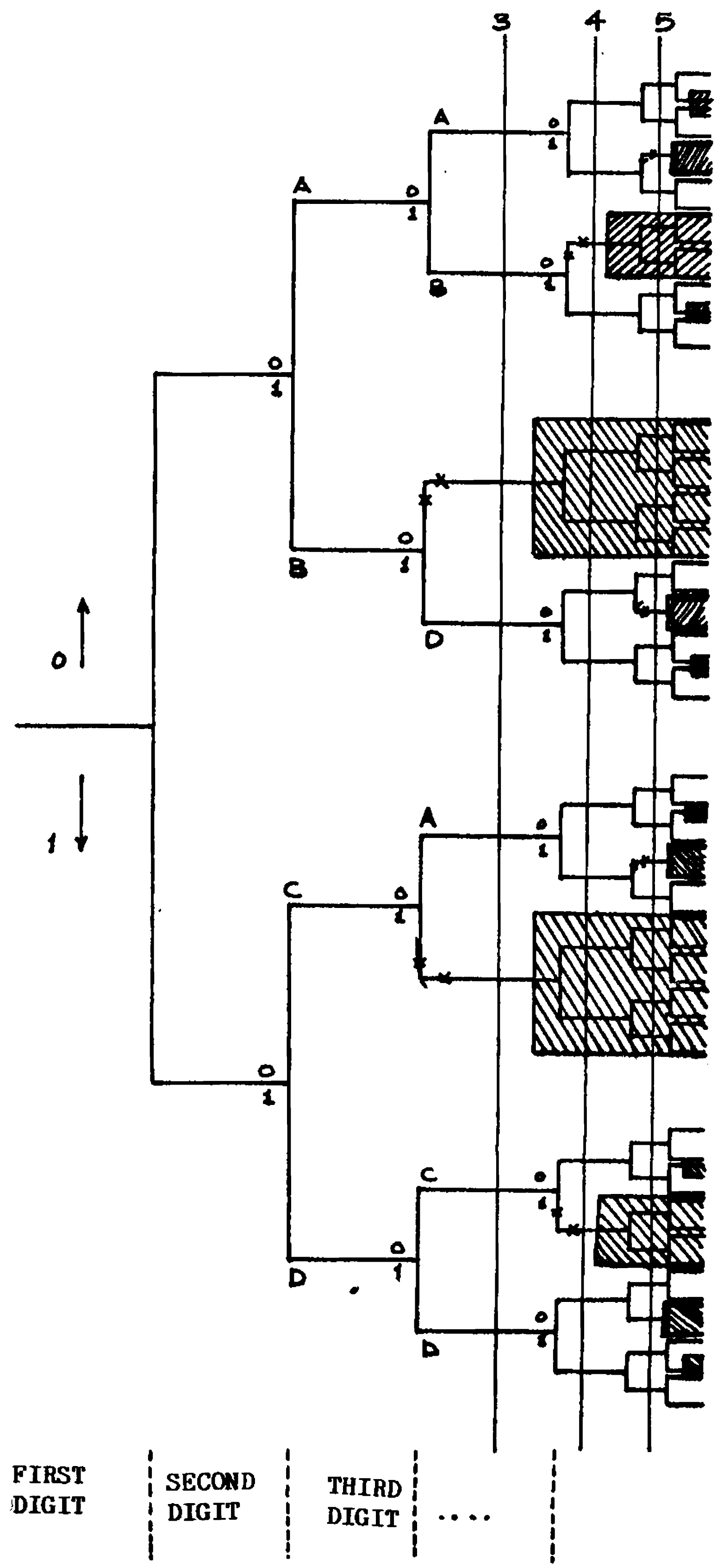
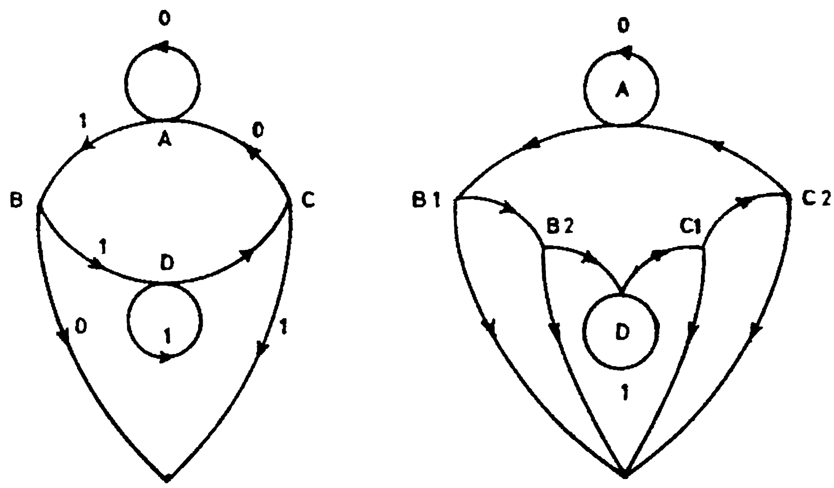


Fig 2.6 TREE STRUCTURE OF A FILTER OF WINDOW SIZE 5.



.SINK
STATE DIAGRAM
WINDOW=3

SINK
STATE DIAGRAM
WINDOW= 5

FIG . 2·7

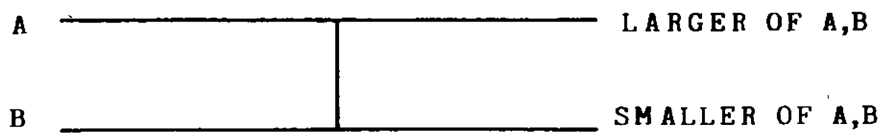
The use of such a tree representation provides a pleasing approach to define the states of the model. However when a signal is quantised to 'q' levels it is not feasible to draw trees for even relatively short signals. Fitch et. al. [25] have developed a general state description for the root signal set without using trees. This model is unrestricted in terms of window size and number of signal quantisation levels. The result is complete and yields an exact system of equations for finding the number of root signals associated with any median filter.

2.3 Methods of median filtering:

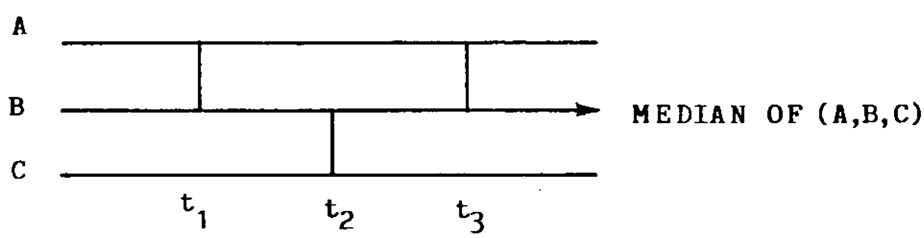
Median filtering gaining momentum in digital signal processing is not only due to its ability to preserve sharp edges while acting as smoother but its simplicity in realisation. Since the introduction of 'Median Filtering' by Tukey, several on-line and off-line median filtering methods have been proposed both inhardware and software. These methods may be classified as:

1. Selection network method
2. Radix method
3. Histogram method

The selection networks [10,11] are a special class of sorting networks and are arrangements of comparators for finding the largest element of a given set. An efficient hardware realisation technique developed by Shamos [10] is shown in fig.2.8. Let the MF window size be 3. First A and B are compared and the larger of 'A and B' placed on the top line, while the smaller of 'A and B' is placed on the middle line. Next the smaller of 'A and B is compared with C'.Again the larger value is



(a)



(b)

FIG. 2.8.

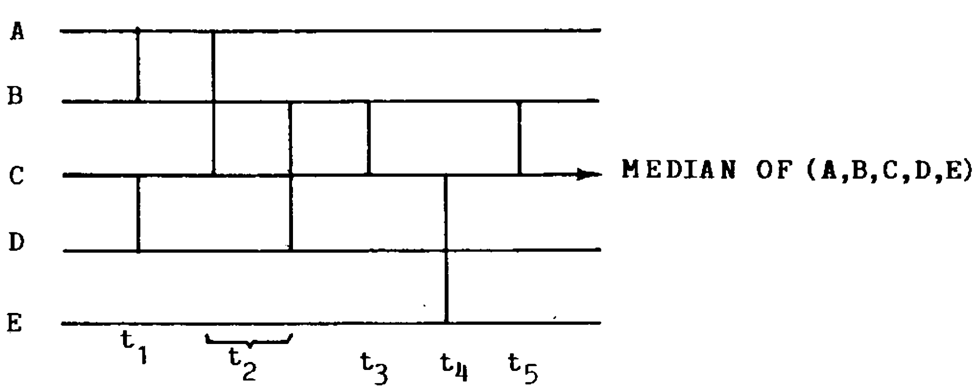


FIG. 2.9.

placed on the top line of the two being compared. The last comparison is made between the larger of 'A and B' and the larger of 'C and smaller of A and B'.This median operator requires three comparisons and the median sample always appear on the middle line. This approach can be applied to five point window as shown in fig.2.9. This needs seven comparisons to achieve the median with an overall time delay of five samples. When the size of window exceeds 7 this method becomes complex. Also a structure either in terms of minimum comparison or in terms of minimum delay is not available.

The Radix method of Ataman et. al. [14] is based on the binary representation of the elements in set $\{ x_i \}$ and subsequently recognizing the various bits in the word. Let the binary representation of x_i be $(b_1^i \; b_2^i \; .... \; b_L^i)$ and that of median $\phi(x_i)$ be $(\mu_1, \mu_2 \cdots\cdots\cdots \mu_L)$. The algorithm for determining the median starts by dividing the elements of the set $(x_i, 1 \leq i \leq n)$ into two groups. The first group contains all those elements of the set for which b_1^i is one and the second group the the rest of the elements. If a majority of b_1^i, $i = 1, \; ... n$ are equal to 1(Ø), then $\mu_1 = 1(o)$. This determines the first bit (most significant bit) of the median. If $\mu_1 = 1$, then the median ϕ is an element of the set of those elements which have $b_1^i = 1$, say set $S(b_1^i = 1)$. Let the cardinality of this set be C. If all

$b_1^i = 1$ i.e. $S(b_1^i = 1) = \{x_i, 1 \leq i \leq n\}$ then obviously ϕ is also the median of the set $S(b_1^i = 1)$. If some $b_1^i = \emptyset$, then ϕ is the (K+1)th largest element of $S(b_1^i = 1)$ when $2K+1 = n$. If a majority of $b_1^i = \emptyset$, then the median ϕ is an element of the set $S(b_1^i = \emptyset)$ and is the ((K+1)-C)th largest element of this set. Assuming without loss of generality that the median ϕ is an element of the set $S(b_1^i = 1)$ the algorithm proceeds by operating on set $S(b_1^i = 1)$ and subdividing the set based on b_2^i = being 1 or $\emptyset$. The search procedure leads to a tree structure.

The method given by Huang et. al [11] is based on the histogram of elements in the set $\{x_i, i=1, 2. .... n\}$. Let $h_n(a)$ be the number of elements such that $x_i = a$ and let $\beta_n(\omega)$ be

$$\beta_n(\omega) = \sum_{a=0}^{\omega} h_n(a)$$

By definition $\beta_n(\phi . x_n) = \frac{n+1}{2}$. If $\beta_n(\omega) = \frac{n+1}{2}$ then ω is the median. If $\beta_n(\omega)$ is greater or less than $\frac{n+1}{2}$, then ω is decremented or incremented by 1 until $\beta_n(\omega) = \frac{n+1}{2}$.

The first two methods are suitable for on-line filtering while the last one is essentially for off-line filtering. It can be seen from fig. 2.8 the minimum number of comparisons for the selection network method for a 3 point and 5 point window are 3 and 7 respectively. However, the number of comparisons and the complexity increase rapidly with increasing window size. Further work done in this area is discussed in detail in Chapter VI.

Speed and complexity of the radix and the histogram methods do not depend on window size but on the word-length L of each sample. Worst case running time of the histogram method grows exponentially with L, while that of the radix method is proportional to L. These two methods are suitable for software realisation.

2.4 Median filtering applications:

The median filtering has been applied to speech signals and image processing. As indicated in 2.1 median filtering preserves sharp discontinuities in the signal and hence may be used for applications where the signal in addition to having high frequency contents is corrupted by high frequency noise. However, MF may fail to provide sufficient smoothing of undesirable noise like components. To overcome this Rabiner at. al. [2] suggested the use of a combination of linear and median smoothers. The smoothing algorithm arrangement is shown in fig.2.10. The input can be written as $x_{(n)} = s\{x_{(n)}\} + r\{x_{(n)}\}$ where s indicates the smooth part and r the rough part of $x_{(n)}$. Then with reference to the fig.2.10 the output $Y_{(n)} = s\{x_{(n)}\}$ and $Z_{(n)} = x_{(n)} - y_{(n)} = r(x_{(n)})$. Thus additional

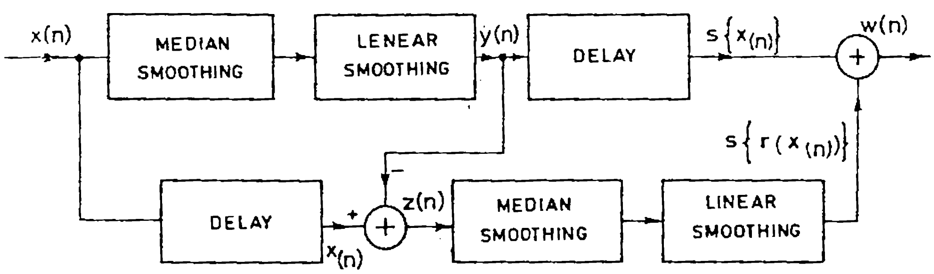


Fig 2.10 BLOCK DIAGRAM OF SMOOTHER AND DOUBLE SMOOTHING ALGORITHMS

smoothing of $Z_{(n)}$ yields a correction term which is added back to $y_{(n)}$ to give $w_{(n)}$, the second approximation to $s(x_{(n)})$. Thus $W_{(n)}$ satisfies the relation

$$W_{(n)} = s\{x_{(n)}\} + s\{r(x_{(n)})\} \quad \ldots. (2.12)$$

The delays shown in the fig.2.10 are necessary to compensate for the inherent delays in filtering. Rabiner et al. suggest the use of a 3 point median filter and a 3 point Hanning window. They have also compared the performance of linear smoothers, median smoothers and a combination of median and linear smoothers.

In speech processing, measurement and processing errors introduce single or double point discontinuities and as such median filters are particularly suited for the removal of such errors. Rabiner has demonstrated the superiority of a combination of smoothers in the processing of speech intensity data and pitch period contours. Further work in this area is discussed in Chapter VI. Jayant [4] has shown the use of MF in communication. Bit errors in DPCM cause propagating distortion in the decoded waveform. The error is essentially impulsive and a median filter squelches the impulse without smearing the speech waveform. Steele and Goodman [5] have further explored the application of MF in smoothing transmission error in linear PCM.

The application of median filtering to image processing is discussed by several authors [3,4,7,23]. Narendra [16] has considered processing images by separable filters rather than two dimensional filters. He shows that the performance of smoothing of the separable filter is identical to that of a square window filter. He also notes that a separable filter yields a slightly greater variance than its two dimensional MF filter. Further work on median interpolation for images are discussed in detail in Chapter IV. The MF application in picture processing for feature extraction is dealt in Chapter VI.

2.5 Conclusions:

In digital signal processing, median filtering offers an alternative to linear filters in some special areas. If the signal contains high frequency component then MF performs better than linear filters in preserving discontinuities in the input. This property of MF is found attractive in speech and image processing. Speech and image contain a large amount of data. Smoothing of such data in real time requires more processing time. Median filter do not involve any arithmetic operation and hence have a large potential for high speed application.

Chapter - III

STRUCTURE AND ANALYSIS

Digital filters are characterised by difference equation. Various structures and response of the filter structures (frequency and phase) can be derived from these difference equations. Unfortunately,being non-linear, median filters are not amenable to this standard analysis technique. When a random signal is used as input, the MF makes the output distribution difficult to calculate and comprehend [19]. It is difficult, if not impossible, to find the output sequence of a MF for any given random signal without actually performing the median operation. In the succeeding part of this chapter a new method of median filter characterisation through matrix operaters is introduced. From this a new parameter 'COLUMN SUM' is extracted. Several features of the signal are deduced from the column sum. Before describing certain structural properties, Median Filtering Window selection and their effect on deterministic signals are discussed.

3.1 MF window selection:

Monotonic sequences and neighborhoods are invariant under median operator of any window size. On the other hand, a sequence containing only oscillations (bivalued) is altered by a median filter reducing the number of oscillation with each pass until tne two neighborhoods are fully extended on either side. Median filtering is of very little use for smoothing these signals, whatever be the window size. However, if we consider an input sequence with spiky noise or 'salt and pepper noise', median filtering is quite efficient. Unlike linear filters, median filter retains sharp discontinuities normally without smearing. If at all smearing occurs, it is a function of window size and also of duration of discontinuity in the input sequence.

To illustrate the window size effect, let us consider an input sequence shown in fig. 3.1

```
           x x x                   x x x x x
   x x x x       x x x x x x x               x x x x x
```

Fig. 3.1

The input samples exhibits sharp discontinuties at sample numbers n=5, 8, 15 and 20. When this signal is passed through MF filters of window 3 and 5, the output is unaltered. The sharp discontinuities at n=5 and n=8 vanish when the sequence is passed through 7 point window. The discontinuities at n=15 and n=20 are unaltered. The discontinuity samples between 5 and 8 form a neighborhood ((K+1) samples) for a window of size 5 i.e. K=2. When the window size is increased i.e. K=3, this portion is no longer a neighborhood since it is less than 4 samples. Similarly, sample numbers between 14 and 20 form a neighborhood upto window sizes of 9 (K=4). This portion will be smoothed out by a median filtering window 13 and above ($K \geq 5$). Thus while choosing a median filter one needs to decide the discontinuities to be preserved for a given input sequence. Thus the window size and the input sampling rate determine the removal of impulses in the input sample sequence. In general, if the discontinuities of m samples and above are to be preserved in MF (the median filtering), K must be less than m.

Effect of window size on deterministic signals:

Let {x(n)} be the signal sample sequence of a triangular waveform shown in fig. 3.2. The signal is sampled to get two samples in a period. The effect of median filter for 3 point window (K=1) can be seen in fig. 3.2(b). The MF output is the same as the input signal except for one sample delay (180 degree phase delay). The 5 point median filter on the same input sequence does not change the signal except for introducing a delay of one period. (Fig. 3.2(c)). This delay is obvious because of start and end effect of MF. To see the effect of number of samples per cycle, let the number of samples per cycle be increased to say 4 samples. Now one can see the effect of median filtering for K=1 and K=2 window size. The input-output waveforms are shown in fig. 3.3. The 3 point median filter output has completely destroyed the periodicity of the input sequence. The output is just a D.C. The MF output for 5 point window also produces a similar DC output.

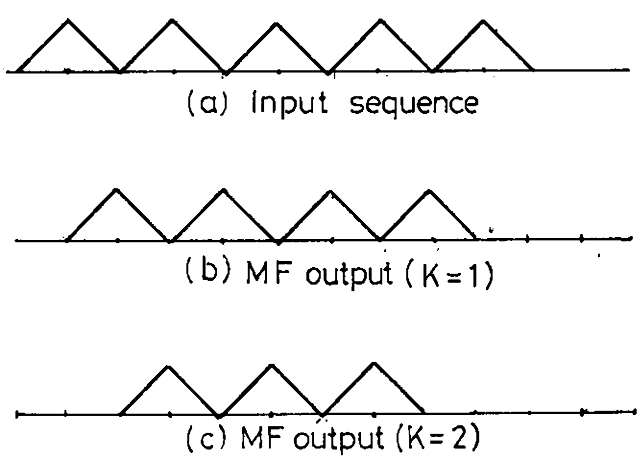
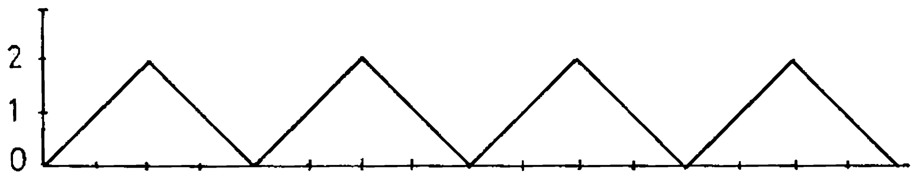
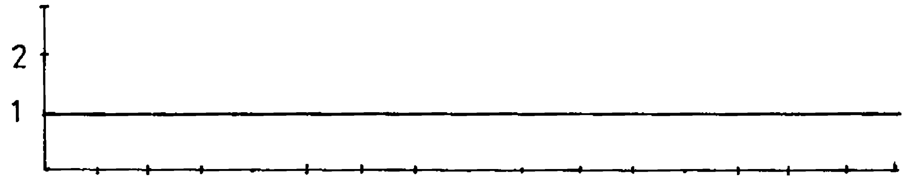


Fig. 3.2



(a) Input sequence



(b) Output sequence

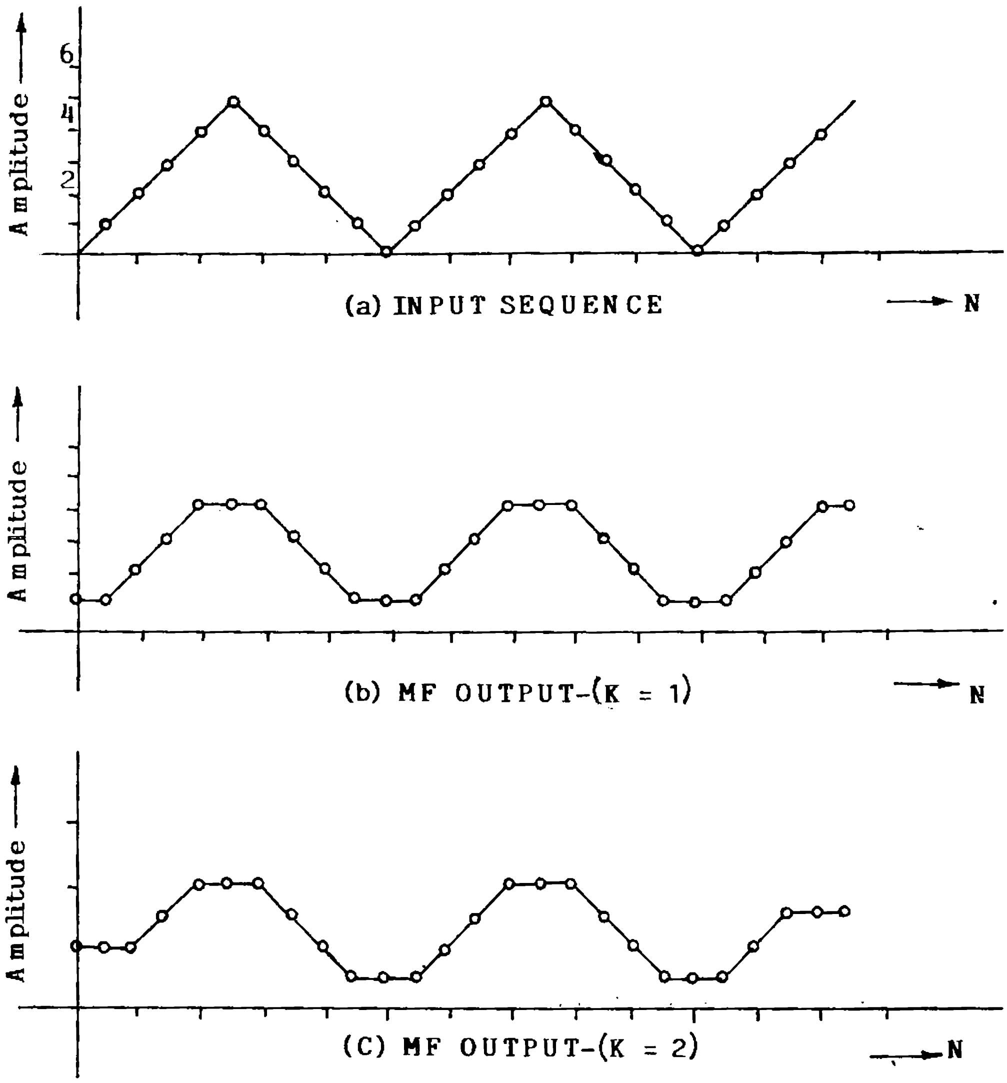
Fig. 3.3

Let the sampling frequency be further increased as for the input sequence shown in fig. 3.4. Here the number of samples in a half period T/2 >(2K+1) for both 3 point and 5 point window. The effect of median filter is shown in fig. 3.4(b) and 3.4(c). In both cases, the slope change over point or signal trend change over point is flattened and a median filter acts as a limiter.

The conclusion from the preceeding discussion is that the input sampling frequency determines the MF output for a fixed window size. The MF smooths out the impulse, slope change over points and oscillation by extending neighborhoods. Finally the extent of smoothing is determined by the window size.

3.2 MF viewed as Transformation:

Locally monotonic functions are invariant under median filtering. Due to this reason they have an important place in median filtering. Tyan [8] was the first to point out their importance and to deduce many important theorems about their properties under median filtering. In the present discussion, a locally monotonic (LOMO) sequence is used to define order



Amplitude
6
4
2
(a) INPUT SEQUENCE
N
Amplitude
(b) MF OUTPUT-(K = 1)
N
Amplitude
(C) MF OUTPUT-(K = 2)
N

FIG. 3.4.

preserving/order reversing transformations. Such transformations are useful in exploring some of the properties of a median.

3.2.1 Order-preserving and order-reversing transformation

Let $Z_i = T\{x_i\}$ where T is a transformation. If for every $x_i \le x_j$ $Z_i \le Z_j$ then T is an order preserving transformation. If on the other hand, if $Z_i > Z_j$ for $x_i < x_j$ then T is an order reversing transformation. Let Ω be the set of all order-preserving or order-reversing transformations. Then $T \in \Omega$

Theorem 3.1

Running median Φ and order-preserving transformation T are commutative

$$\text{i.e.} \quad \Phi\{T\{x_n\}\} = T\{\Phi\{x_n\}\} \quad .... \qquad (3.1)$$

Proof:

Let T be order-preserving and $\{Z_n\} = T\{x_n\}$ and $Y = \Phi\{x_r\}$
Let W_r be a (2K+1) window at the r-th position in the RM. Then by the definition of the median, there are K elements greater than or equal to Y in the window W_r.
Let $Y = X_j$ Then $Z_j = \Phi\{Z_r\}$

where $Z_r = T\{x_r\}$ $x_r \in W_r$ as by the definition of T, if $x_i \le x_j$ then $Z_i \le Z_j$. Hence we have

$$Z_j = \Phi\{Z_i\} = \Phi\{T\{x_i\}\} \quad x_i \in W_i \qquad .... \qquad (3.2)$$

and by definition Z_n

$$Z_j = T\{x_j\} = T\{\Phi\{x_r\}\}, \quad x_r \in W_r \qquad \ldots\ldots\ldots \qquad (3.3)$$

Equations (3.1), (3.2) and (3.3) hold for all i. The inequalities in the above proof will be reversed for order reversing transformations.

Monotonic sequence being a fixed point (root) of a median we have:

Theorem 3.2

The running median operator is an order preserving transformation on any monotonic sequence.

Proof:

Let T be an operator on x_n for order transformation

$$Z_n = T\{x_n)$$

Z_n is the transformed sequence and is said to be order preserving only when

$$Z_n < Z_{n+1} < \ldots\ldots < Z_{n+i} \qquad i=\emptyset,1,2,3 \ldots\ldots$$

Let Φ be the running median operator for a window W on the input sequence.

$$\Phi\{Z_n\} = \Phi\{T\{x_n\}\}$$

By the definition of median, it is $(K+1)^{th}$ largest or smallest element of Z_n. For a monotonic sequence the output is the same as the input sequence. Therefore the median filter output of the order-transformed sequence is always the central sample. In running median, the window W is sliding on the input, and the output follows the input. Thus the RM is an order preserving transformation for monotonic sequences.

Theorem 3.3

Any sequence transformed to order preserving or order reversing transformation is also a root sequence of the median filter.

Proof:

Let $Z_n = T\{x_n\}$ and $\Phi\{Z_n\} = R_n$ where R_n is a root signal.

From theorem 3.2 Φ is an order-preserving operator.

Therefore

$$R_n = Z_n$$

as all the sequences undergoing order-preserving transformation are root signals for running median.

From the above theorems and the discussions the following can be concluded. With $T \in \Omega$,

(a) If T_1 and $T_2 \in \Omega$, then so are $T_1 . T_2$ and $T_2 . T_1$,

(b) If $\{x_n\}$ is LOMO(m), then so is $Z_n = T\{x_n\}$.

It is possible to generate a new LOMO(m) sequence from a known LOMO(m) using the statements (a) and (b). Some examples are listed as follows:

(i) $Z_n = T\{x_n\} = a x_n + b$ where a and b are real constants, with $a \neq 0$.

(ii) $Z_n = T\{x_n\} = x_n^p$ where $p \neq 0$ and either all $x_n > 0$ or all $x_n < 0$.

(iii) Let both x_n and y_n be monotonic sequences with the same trend then $Z_n = a x_n + b y_n$ is also a monotonic sequence of the same trend when $a,b > o$ and of opposite trend when $a,b < \emptyset$

(iv) $Z_n = T\{x_n\} = a_1 x_n^{p1} + a_2 x_n^{p2} + \ldots + a_r x_n^{pr} + c$ where all p_i and a_i have the same sign (a_i might be of different sign from p_i). Also not all a_i can be zero and all x_n are either non negative or non-positive.

3.3 Characterisation of Median Filtering:

Though application of median filtering in signal processing has received considerable attention, a rigorous method of characterising 'median' is still not available. Here an attempt is made to characterise median by a 'Matrix' operation. In the sequel a three point median (K=1) is considered. By definition, median is the middle value of the ranked three input samples. Ranking is possible only when all the three input samples are available. A delay of 2 samples is unavoidable with a window size 3 (K=1) median filter. The median output corresponds to $(K+1)^{th}$ sample. Therefore the running median output at i may have a maximum of $\pm K$ sample displacement. The structure of a median filter using delays and coefficients is shown in fig. 3.5, a structure similar to that of a digital filter. The salient features of MF vis-a-vis those of digital filters are :

(i) MF filter response delay is 2K unit samples for a (2K+1) window whereas digital filter responds for every bounded input.
(ii) The coefficients a, b and c in MF are either 1 or 0 . Further only one of them can be non-zero at any sample instant, whereas in digital filters these coefficients are real with their values determined by the desired filter response.
(iii) The signal flow graph of median filter is the same as that of digital filter with the structure of fig. 3.5.

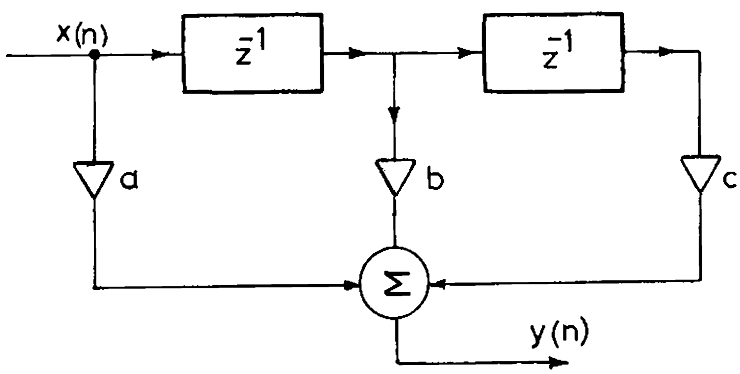


Fig. 3.5

The input-output relationship of a MF can now be written as

$$y_{(n)} = a\,x_{(n)} + b\,x_{(n-1)} + c\,x_{(n-2)} \quad \ldots\ldots \quad (3.4)$$

where $x_{(n)}$ is the input, $y_{(n)}$ the output while a, b and c are co-efficients (either 0 or 1). Equation (3.4) may be written as

$$y(n) = [x(n) \quad x(n-1) \quad x(n-2)] \cdot \begin{bmatrix} a \\ b \\ c \end{bmatrix} \quad \ldots\ldots\ldots\ldots\ldots \quad (3.5)$$

$$n = 1,2 \ldots L$$

At any instant n only one of a, b, c is 1, the other two being zero placing in evidence the median of the specific window. The specific values of a, b, c depend on the signal. For instance for a monotonic sequence 'a' and 'c' are Ø while b=1. This characterisation leads to a possibility of expressing the median filter in terms of linear operation. Equation (3.5) can be generalised to get output matrix $Y_{(n)}$, that is

$$\begin{bmatrix} y_{11} & y_{12} & \cdots & y_1(l-2) \\ y_{21} & y_{22} & \cdots & y_2(l-2) \\ \vdots & \vdots & & \vdots \\ y_{(l-2)1} & y_{(l-2)2} & \cdots & y_{(l-2)(l-2)} \end{bmatrix} = \begin{bmatrix} x_1 & x_2 & x_3 \\ x_2 & x_3 & x_4 \\ \vdots & \vdots & \vdots \\ x_{(l-2)} & x_{(l-1)} & x_{(l)} \end{bmatrix} \begin{bmatrix} a_1 & a_2 \ldots a(l-2) \\ b_1 & b_2 \ldots b(l-2) \\ c_1 & c_2 \ldots c(l-2) \\ & \ldots \end{bmatrix} \quad (3.6)$$

$$\left[Y_{(n)}\right] = \left[X\right] \cdot \left[W\right]$$

where [X] is a (L-2) x 3 matrix. Row of X consists of signal samples that fall in a (2K+1) window. [W] is a 3x(L-2) weight matrix which has only one non zero entry in a column that corresponds to the median sample for the window. The output $Y_{(n)}$ is a (L-2)x(L-2) matrix . Finally the median vector Y is given by

$$Y = \text{Diag}\,[Y_n] \qquad \ldots\ldots \ (3.7)$$

It can be observed that each column of Y_n is the input signal vector in natural order or cyclically rotated a maximum of 2K times. An example is given to illustrate this. (see Fig.3.6)

$\overrightarrow{\boxed{3\quad 5\quad 2}}$ Ø 1 3 2 6 6 4 3 1 2 ... Signal Sequence

Window

$$
\begin{bmatrix}
3 & 5 & 2\\
5 & 2 & 0\\
2 & 0 & 1\\
0 & 1 & 3\\
1 & 3 & 2\\
3 & 2 & 6\\
2 & 6 & 6\\
6 & 6 & 4\\
6 & 4 & 3\\
4 & 3 & 1\\
3 & 1 & 2
\end{bmatrix}
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0\\
0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0\\
0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}
=
\begin{bmatrix}
3 & 5 & 2 & 5 & 2 & 3 & 5 & 3 & 5 & 5 & 2\\
5 & 2 & 0 & 2 & 0 & 5 & 2 & 5 & 2 & 2 & 0\\
2 & 0 & 1 & 0 & 1 & 2 & 0 & 2 & 0 & 0 & 1\\
0 & 1 & 3 & 1 & 3 & 0 & 1 & 0 & 1 & 1 & 3\\
1 & 3 & 2 & 3 & 2 & 1 & 3 & 1 & 3 & 3 & 2\\
3 & 2 & 6 & 2 & 6 & 3 & 2 & 3 & 2 & 2 & 6\\
2 & 6 & 6 & 6 & 6 & 2 & 6 & 2 & 6 & 6 & 6\\
6 & 6 & 4 & 6 & 4 & 6 & 6 & 6 & 6 & 6 & 4\\
6 & 4 & 3 & 4 & 3 & 6 & 4 & 6 & 4 & 4 & 3\\
4 & 3 & 1 & 3 & 1 & 4 & 3 & 4 & 3 & 3 & 1\\
3 & 1 & 2 & 1 & 2 & 3 & 1 & 3 & 1 & 1 & 2
\end{bmatrix}
$$

$$[X] \qquad [W] \qquad = \qquad [Y_n]$$

$$Y = \square\ , 3, 2, 1, 1, 2, 3, 3, 6, 6, 4, 3, 2, \square$$

Fig. 3.6

It may be noted the matrix Y_n is always a square matrix. For certain types of signal Y_n can be directly written. For Root signals, periodic signals and bivalued signals Y_n can be directly written by inspection. When the input signal vector is monotonic, it undergoes one rotation whereas the signal vector containing neighborhood appear without any rotation in the output square matrix Y_n. However, no clear cut rule emerges for writing this for a general signal. The matrix W is modified in order to extract several properties of a median filter.

3.4 Median Matrix

The usefulness of the transformation matrix is improved by modifying W, the weighting matrix. This is achieved by padding the matrix with suitable number of zeros so that it becomes a (L-2K) x L matrix. This matrix is called the Median Matrix M. This M matrix has only one entry of 1 per row. This transformation matrix represents the mapping of the input into output. The median extraction operation using M matrix can be written as

$$Y = M\,\bar{X} \qquad \ldots\ldots \qquad \ldots\ldots \qquad (3.8)$$

where Y is the output column vector and $\bar{X}$ is the input column vector, while M is the Median Matrix. An important parameter of M that can be used to extract some properties of the signal is the 'COLUMN SUM'. The column sum can be defined as the additive value of each column of M matrix. The column sum indicates the input samples that appear at the MF output along with the number of times each sample appears at the output. It can also be used to indicate the trend of a signal The example in Fig.3.7 illustrates the Median matrix and the Column sum. M can be seen to be a banded matrix The column entries of W become the row entries of M along the band shown. It is obvious that each row of this matrix has only one entry of 1 and each column can contain a maximum of three 1s for K=1. Similar matrices can be obtained for other values of K.

3 5 2 Ø 1 3 2 6 6 4 3 1 2 Input sequence

(a)

$$\text{Median Matrix } M = \begin{bmatrix}
1 & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset \\
\emptyset & \emptyset & 1 & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset \\
\emptyset & \emptyset & \emptyset & \emptyset & 1 & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset \\
\emptyset & \emptyset & \emptyset & \emptyset & 1 & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset \\
\emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & 1 & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset \\
\emptyset & \emptyset & \emptyset & \emptyset & \emptyset & 1 & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset \\
\emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & 1 & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset \\
\emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & 1 & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset \\
\emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & 1 & \emptyset & \emptyset & \emptyset \\
\emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & 1 & \emptyset & \emptyset \\
\emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & 1
\end{bmatrix}$$

Column Sum = 1 Ø 1 Ø 2 1 1 2 Ø 1 1 Ø 1

(b)

Fig 3 7 (a) Input sequence (b) Median matrix M and column sum

The sum of the elements in any column S_i and the sum of the column sums CS have many interesting properties. These can be used for extracting signal properties. These properties of the column sums and CS are listed in the following:

(1) For a given input sequence of length L, the summation of the column sums (CS) is equal to (L-2K)

i.e. $CS = \sum_{i=1}^{L} S_i = (L-2K)$ .... (3.9)

where S_i is the column sum of the i^{th} column.

(2) For a given window (2K+1) the column sum S_i lies between Ø and (2K+1)

i.e. $0 \leq S_i \leq (2K+1)$ ..... (3.1Ø)

(3) Sum of any two successive sums cannot exceed (2K+2)

$\emptyset \leq (S_i + S_{i+1}) \leq (2K+2)$ (3.11)

(4) Sum of n successive column sums cannot exceed (2K+n)

i.e. $\sum_{i=1}^{n} S_i \leq (2K+n)$. (3.12)

(5) A given value of column sum S_i repeats a maximum of $[L/S_i]$ times in a sequence of length L where [ ] indicates the integer part.

(6) The number of successive appearances (SR) of a given S_i is given by

$SR = [(2K+1)/(S_i - 1)] - 1$ .... (3.13)

(7) There can not be (2K+1) successive S_i's equal to zero

i.e. $\sum_{i=m}^{2K+m} S_i \neq \emptyset$ .... (3.14)

Proof:

For a sequence length of L, there are (L-2K) window positions each representing a row in M matrix. Each row is extracting one sample from the input sequence as median for that instant i hence there is only one entry of 1 per row. Each column sum S_i therefore maps the number of 1's in that particular column. Thus the number of 1's mapping onto the total column sum is equal to the number of 1's present in the M matrix. As per the definition of M matrix (L-2K) x (L-2K) this number is equal to (L-2K) Here the start and end effect of window are not considered

No. of 1 s in each row = 1.

Total No. 1's in L rows = L

No. of rows contributing to each column sum = 2K+1

Therefore the maximum no. of 1's adding to produce S_i = 2K+1.

No of rows contributing to two successive column sums = 2K +2

Therefore the maximum no of 1's adding to the two successive S_i's = 2K+2

No. of rows contributing to n successive column sums = (2K+n)

Therefore maximum no. of S_i's for n successive column sum = (2K+n)

Thus the properties (1) to (4) are satisfied with the preceeding arguments.

Property (5) can be established by the succeeding argument In the signal sequence, number of segments equal to the window size (2K+1) is L/(2K+1).

From property (2) $S_i \leq (2K+1)$

Therefore the number of times column sum S_i can repeat in the sequence is= L/S_i. Thus the property (5) is satisfied. The property (6) is derived using property (2) and (4).

Given a window (2K+1) at any instant i, the median sample is necessarily at i or $i \pm K$. Therefore the weighting is within the window (2K+1) Hence (2K+1) successive S_i cannot be equal to zero. Thus the property (7) is proved

Some of these properties are trivially obvious while the others are not so obvious. Property (1) merely places in evidence the fact that the input and output sequences are of the same length. Property (2) limits the number of times a particular sample can repeat at the output The maximum of this understandably limited to the window width since a sample goes out of reckoning beyond one window width. Property (3), not so obvious is also a direct consequence of the fact that a sample goes out of reckoning after 1 window width Property (4) is an extension of property (3) to a general case of n columns. This property is useful in determining the number of different patterns possible for the column sums. Property (5) is a consequence of properties (1) through (4) and the total number of times a sample can repeat. Property (6) is the constraint imposed by the properties (2) and (4) .As it is obvious that the median sample must be within the window segment property (7) follows.

From the properties of the CS some useful conclusions can be drawn as to the possible patterns of column sums, or indirectly the pattern of sample repetitions. It can be shown that the number of different column sum patterns for any length can be calculated from these properties. For example, there are 4, 16, 64 and 256 possible combinations of column sums out of which only 4, 13, 39 and 114 are valid for lengths 1, 2, 3 and 4, respectively This may be generalised for a (2K+1) window and n successive columns.

For a given n the valid combinations V are

$V(n) = (2K+2)^n - IC(n)$ where IC is the total invalid combinations. The total number of combinations for a given n is given by $(2K+2)^n$ The invalid combinations are of two types viz (1) invalids due to tree propagation which is equal to (2K+2) IC(n-1) (2) invalids arising out of properties (3) and (7) at the current n $IC_N(n)$. Therefore

$IC(n) = (2K+2)\ IC(n-1) + IC_N(n).$

Another property of the column sum viz. property (7) leads to invalid combinations. From the structure it can be seen that for K=1 this is simply 3^{n-3} for all $n \geq 3$. The valid combinations arising out of the property (4) that is $\sum_{i=1}^{n} S_i \leq 2K+n$ can easily be calculated using combinatorial arithmetic. However, a very interesting recursive relation is exhibited by these numbers. This relationship for K=1 is given by

$IC(n) = 3V\ (n-2)$ for all $n \geq 3$.

That is the invalid combinations for a given n is (2K+1) times the valid combinations for the (n-2) columns. Or, in general for a (2K+1) window the number of invalid combinations at the n^{th}

column is given by

$$IV = (2K+1)^{(2K-1)} \cdot V(n-2K) \qquad \text{for all } n \geq 2K+1.$$

The result is tabulated in Table III.1 for 3 point and 5 point windows A tree structure can be defined for the propagation paths of the column sum from these column sum properties. This will be discussed in the Section 3.6.

Table III 1

WindowSize = 3				Window size = 5			
Column (n)	Total Combin.	valid combin.	Invalid combin.	Column (n)	Total combin.	Valid combin.	Invalid combin.
1	4	4	0	1	6	6	0
2	16	13	3	2	36	26	10
3	64	39	25	3	216	100	116
4	256	114	142	4	1296	364	932

3.4.1 **Signal properties from column sum:**

Several properties of the signal can be deduced from the column sum. Once the input sequence and K are specified, the S_is have a structure. The column sum can therefore be used to find some properties of signals

1. The pattern of the column sum indicates the trend of the signal

2. A periodic column sum pattern indicates the presence of a periodic signal. This is shown in fig. 3.8. It may be noted here that the period is half that of the input.

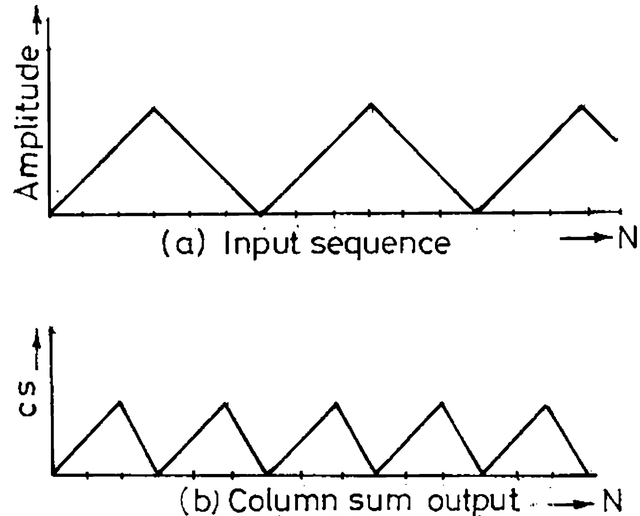


Fig.3.8 (K=1)

3. Trend change over points are indicated in the column sum. $S_i = 0$ preceeded by a non zero digit indicates the presence of maximum or minimum in the signal This is evident from fig.3.8.

4. The column sum pattern may change with successive passes for non root signals. However, for a root it does not change.

A few examples are presented here to illustrate the applicability and usefulness of these properties.

Case 1: A monotonic sequence.

Let x(n) = 0, 1, 3, 5, 6, 7, 9, 10, 11, 12 and

let K=1. The median matrix M can be written as

$$[M] = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

and $[S_i] = 0\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 0$

It may be noted that the first and last S_i's are 0 while the rest are 1's. This is the column sum pattern for all monotonic sequences

Case II: Bivalued signal

Let x(n) = 3, 7, 3, 7, 3, 7, 3, 7, 3, 7 and K=1. Its median matrix and the corresponding S_i's are

$$[M] = \begin{bmatrix} 1 & Ø & Ø & Ø & Ø & Ø & Ø & Ø & Ø & Ø \\ Ø & 1 & Ø & Ø & Ø & Ø & Ø & Ø & Ø & Ø \\ Ø & Ø & 1 & Ø & Ø & Ø & Ø & Ø & Ø & Ø \\ Ø & Ø & Ø & 1 & Ø & Ø & Ø & Ø & Ø & Ø \\ Ø & Ø & Ø & Ø & 1 & Ø & Ø & Ø & Ø & Ø \\ Ø & Ø & Ø & Ø & Ø & 1 & Ø & Ø & Ø & Ø \\ Ø & Ø & Ø & Ø & Ø & Ø & 1 & Ø & Ø & Ø \\ Ø & Ø & Ø & Ø & Ø & Ø & Ø & 1 & Ø & Ø \end{bmatrix}$$

$$[S_i] = 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ Ø\ Ø$$

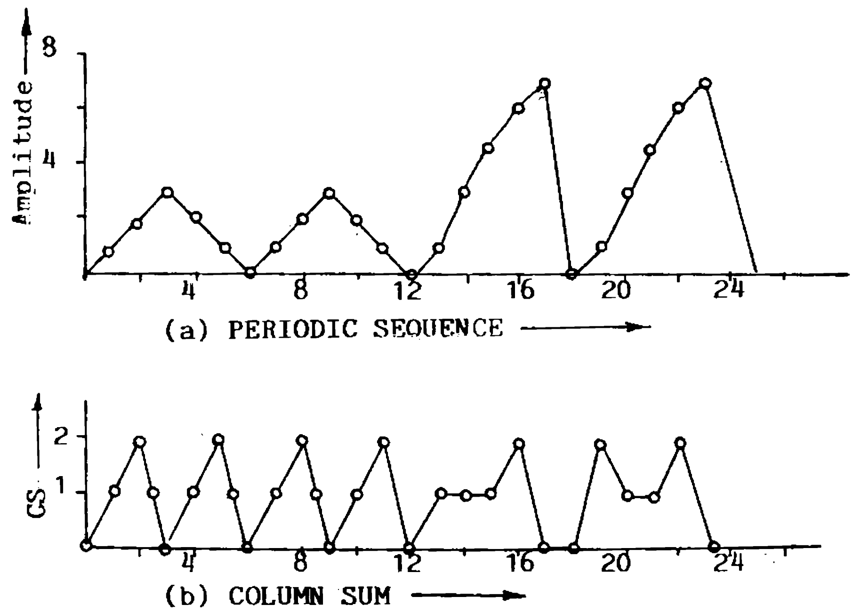
Thus it is a sequence of 1's followed by a pair of Ø's. All bivalued sequences show this pattern on the first pass because the weighting coefficient a=1 and b=c=Ø for all input signal vector X as per the equation (3.5).

Case III: Periodic sequence:

Let x(n) = Ø,1,2,3,2,1,Ø,1,2,3,2,1,Ø,1,3,5,6,7,Ø,1,3,5,6,7,Ø and K=1. This signal sequence shows two periodic segments. One of them is symmetric around its half period and the other is not. The column sum obtained from the median matrix M is

$[S_i]$ = Ø,1,2,Ø,1,2,Ø,1,2,Ø,1,2,Ø,1,1,1,2,Ø,Ø,2,1,1,2,Ø,Ø

It is clear that the column sum $[S_i]$ is also periodic. When the sequence has symmetry around its half period, the column sum periodicity is twice that of the input sequence whereas for others it is the same as that of the input sequence. Further the column sum sequence at which S_i = Ø is an indication of signal maxima or minima which is being flatened in median filtering. This is evident in Fig.3.9.



8
4
Amplitude
4
8
12
16
20
24
(a) PERIODIC SEQUENCE
2
1
CS
4
8
12
16
20
24
(b) COLUMN SUM

FIG. 3.9.

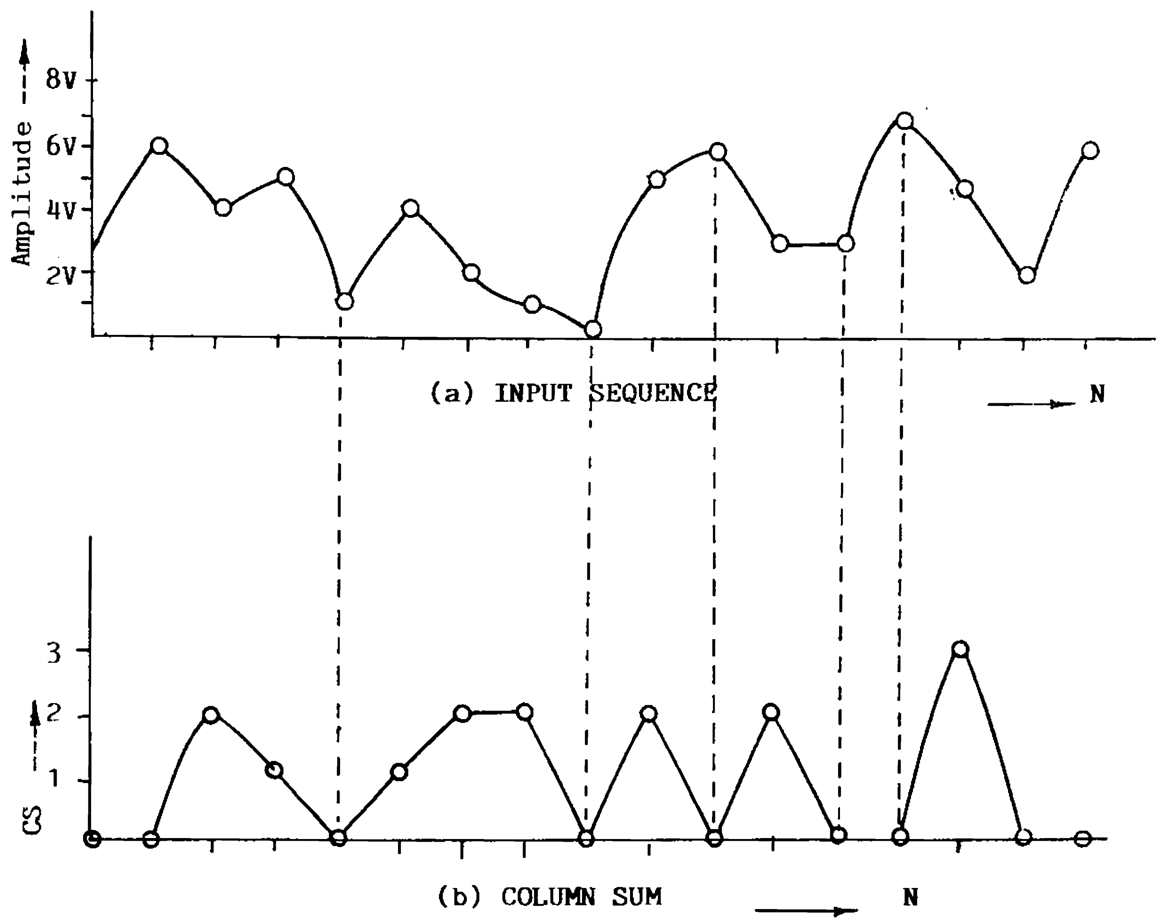


FIG. 3.10.

Case IV: A random sequence:

Consider a random sequence x(n). Let K=1.

x(n) = 3,6,4 5 1 4 2 1 0 5,6 3,3 7 5 2,6

Median output

{Y} = $\boxed{4}$, 4,5,4,4,2,2,1,1,5,5,3,3,5,5,5, $\boxed{5}$

The column sum can be obtained from median matrix as

0 0 2,1,0 1,2,2,0,2,0,2,0,0,3,0,0

On examining it can be seen that S_i pattern is also random. The zeros in the column is invariably a trend changeover point (maximum/minimum) in the input sequence. This is clear from fig.3.10.

3.5 Root Analysis

An input sequence invariant under median filtering with a given K is called root sequence. Tyan has grouped the input sequence into two groups Root sequences such as (a) monotonic function (b) step function (c) stair case signals etc. are grouped as Type I signals. Median filtering is of very little use for a signal containing only oscillations between two levels. Such signals are called bivalued signals and grouped as Type II. Gallagher and Wise have not given the exact number of passes required for any given sequence to reach a root (see Chapter II). They have stated only the maximum number of passes to arrive at a root signal. However, it is possible to arrive at the exact number of passes by investigating the structure of the input sequence. It is to be recalled from the discussion in Chapter II that median filtering reduces the oscillation from both ends The number of oscillations in the sequence is the one which primarily decides the number of passes to reach a root sequence In other words it depends on the number of times the trend (slope) changes in the sequence. Let this be T. Then the number of passes to reach at a root sequence is given by

$$R = \left[\frac{T}{2K} \right] \qquad \text{.......... (3.15)}$$

where [ ] indicates the integer part.

Proof:

Case I: No neighborhood. Let T be number of trend change over points in the input sequence and let there be no neighborhood. For each pass of the signal the MF filter reduces the trends by 2K Then the number of trend changeovers after first pass T1 = T-2K. After second pass T2 = T1 - 2K or = T - 4K.

Let the trend changeover T becomes zero after n passes

$$T_n = T - n\ 2K$$

Equating $T_n = \emptyset$ we get n = T/2K.

i.e. the No. of passes to reach a root sequence R

$$R = T/2K$$

Case II: The input sequence consists of neighborhood, monotonic functions and oscillation.

Let T be the number of trend changeovers and P be the number of neighborhoods and monotonic sequences. Though neighborhoods and monotonic sequences are invariant signals, their presence increases the number of trend changeover points in the sequence. When there are P neighborhoods/monotonic sequences, the trend changeover by this neighborhood can be (P-1). Therefore equation (3.15) becomes

$$R = \left[\frac{T - (P-1)}{2K} \right] \qquad \ldots\ldots\ldots \quad (3.15(a))$$

In the example given in Fig.3.11 the number of changes in trend T=11 The number of neighborhoods P=4 for window K=1. Then the number of passes

$$R = \left[\frac{11 - (4-1)}{2} \right] = 4$$

Ø Ø 1 1 Ø Ø 1 1 Ø 1 Ø 1 Ø 1 Ø 1

(a) Input sequence

Ø Ø 1 1 Ø Ø 1 1 1 1 1 1 1 1 1 1

(b) Root sequence after IV passes for K=1.

Ø Ø Ø Ø Ø 1 1 1 1 1 1 1 1 1 1

(c) Root sequence after II passes for K=2.

Fig.3.11

For the same input sequence, the neighborhoods for K=1 have become oscillation to K=2. The trend changeover points in the input sequence remains unchanged for any window size. Therefore using equation (3.15), the number of passes required for the sequence to converge to a root is

$$R = \left[\frac{11}{4} \right] = 2$$

Thus the equations (3.15) and (3.15(a)) give the exact number of passes to arrive at a root using only the number of trend changeover points and neighborhoods for a given K.

It is easy to see that the following statements regarding root sequence are true.

Statement 1: The algebraic sum of two root sequences of window K is also a root for the same window provided the roots are of the same trend.

Example: Let $y_{1(n)}$ and $Y_{2(n)}$ be two monotonically increasing independent root sequence for K=1.

$Y_{1(n)}$ = 1 5 7 11, 12, 15, 19, 20

$Y_{2(n)}$ = 4, 5, 8, 11, 12, 15, 16, 17

Let $Y_{(n)}$ be the algebaric sum of $Y_{1(n)}$ and $Y_{2(n)}$

i.e. $Y_{(n)}$ = 5, 10, 15, 22, 24, 30, 35, 37

$\Phi[Y_n]$ = $\square$, 10, 15, 22, 24, 30, 35, $\square$ for K=1.

Thus the algebraic sum of two monotonic sequences of the same trend is also a root/monotonic.

Statement 2:

Concatenation of two independent roots for a given window yields a root sequence after the second pass through the same MF, independent of the trends of the original sequences.

Example: Let $Y_{(n)}$ be the concatenated sequence of $Y_{1(n)}$ and $Y_{2(n)}$ which are independent root sequences for MF window K=1.

$Y_{1(n)}$ = 1,2,4,5,8,9,10,12

$Y_{2(n)}$ = 9,8,7,5,3,2,1,0

where $Y_{1(n)}$ and $Y_{2(n)}$ are monotonic sequences but of opposite trend.

$Y_{(n)} = 1,2,4,5,8,9,10, 12, 9,8,7,5,3,2,1,0$

The MF output of window K=1 after the first pass is given by

$\phi\{Y_{(n)}\} = \square,2,4,5.8,9,10,10,9,8,7,5,3,2,1,\square$

The second pass output

$= \square,2,4,5,8,9,,10,10,9,8,7,5,3,2,1,\square$

This is a root sequence.

Root properties do not change for some of the arithmetic operations. The following are valid arithmetic operations on root sequences

(i) If $\{Y_n\}$ is a root $\alpha\{Y_{(n)}\}$ is also a root where α is a constant.

(ii) If $Y_{1(n)}$ and $Y_{2(n)}$ are root sequences of the same trend then a linear combination $\alpha\{Y_n\} + \beta\{Y_{2(n)}\}$ is also a root, where α and β are constants with the same sign.

(iii) The product or division of two root sequences of corresponding samples is also a root provided the trends are the same for both.

(iv) Unlike arithmetic operation logical operation on root sequences do not yield a root

3.6 Tree structure of column sum:

As per the properties of the column sum, the maximum number of possible combinations and valid combinations for column numbers 1 through 4 is listed in Table III.1 for window K=1 and K=2. It may be noticed that the number of valid combinations increases rapidly as the number n increases. This growth may be represented in a tree structure. A tree structure is developed using the column sum properties. This is shown in fig 3.12 for K=1. The branches which are not permitted are marked X in the tree diagram

It may be recalled from the discussions in the preceeding section 3.5 that it is possible to deduce some of the signal structure like monotonic, neighborhood, periodicity of a signal, maximum/minimum etc. It may also be noted that the column sum values take a definite pattern for invariant signals and do not depend on the input sequence quantisation level. This method gives an elegant tree structure. Fitch et. al [25] worked on the actual signal to evaluate the root paths. The number of root paths are very high for a given length of sequence n though the input sequence structure are the same. To illustrate this let us consider a sequence length n=2. Arce has shown that there are 4 possible roots for binary signal (ØØ, Ø1, 1Ø, 11). Fitch has shown that there are about 16 possible root paths for 4 level signal (ØØ, Ø1, Ø2, Ø3, 1Ø, 11, 12, 13, 2Ø, 21, 22, 23, 3Ø, 31, 32, 33). In the column sum method, only the number of root structures are identified rather than its actual values. Hence there are only three possible root paths viz. Ø1, 1Ø, 11. The root paths corresponding to binary and 4 level signals are:

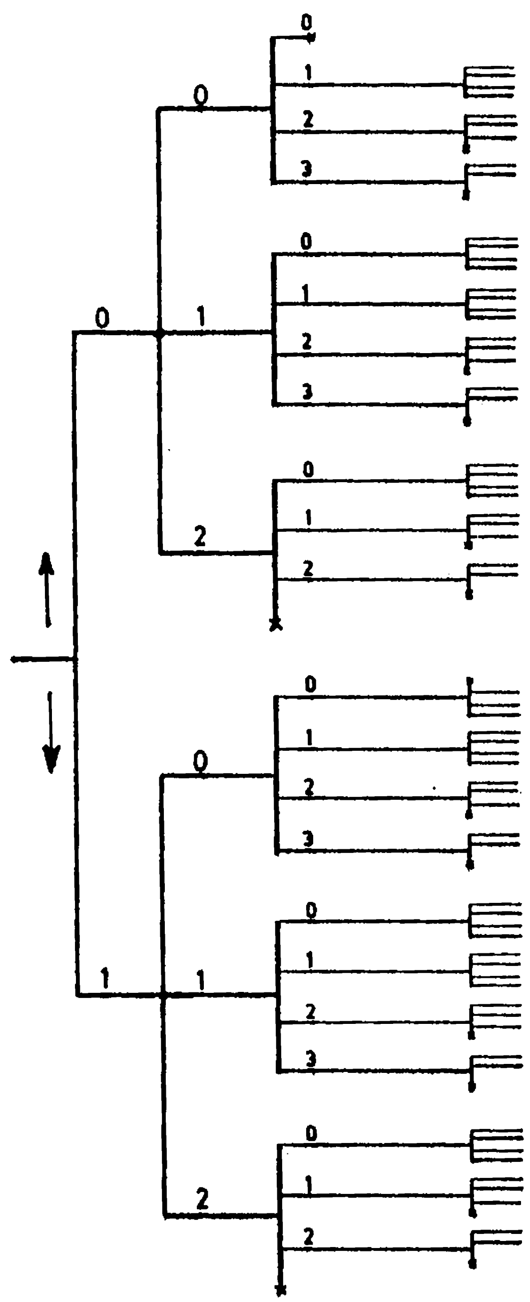


Fig.3.12 Tree diagram (K=1)

Root paths (CS method)	Ø1	1Ø	11
Binary signal	Ø1	1Ø	ØØ, 11
4 level signal	Ø1,Ø2,Ø3 12,13,23	1Ø,2Ø,21 3Ø,31,32	ØØ,11, 22,33

Similarly this can be extended for other values of n.

To generate a root signal,the initial column sum value can be either Ø or 1 for i=1 due to start effect and also the weighting coefficients of the signals can be either 1ØØ or Ø1Ø. For i=2 the root path generation is decided by the first digit of S_i. When it is Ø the second digit can take only value 1 (Ø1) since the rest of the weightings leads to non root paths. If the first digit is 1, the second may take either Ø or 1 (1Ø,11). The start effect continues to impose restriction on S_i until i=2K for (2K+1) window Thus there are three possible combination for i=2.

The root path generation for i=3 can be identified by looking into the first two S_i's. The third S_i corresponds to a monotonic sequence whereas Ø12 corresponds to a combination of monotonic and neighborhood signal. When the S_i is 1Ø, the third may take 1 (1Ø1) for root path. This corresponds to a input sequence havingneighborhood followed by monotonic sequence. If the first two digit of S_i s are 11, the third S_i may be either Ø or 1 i.e. 111 or 11Ø. The S_i (111) corresponds to a neighborhood and (11Ø) may correspond to a neighborhood followed by a monotonic sequence. Thus there are 5 possible root paths for i=3. Thus the root path can be traced in the tree diagram using the weighting matrix corresponding to the root signals.

The generation of root signal paths for any length of signal can also be represented as state diagram as shown in fig.3.13. The states are defined as follows:

State A: Two successive digits with values 1 (1,1). The next digit can take value either Ø or 1.

State B: Two successive digits with value 1 (1,1). The next digit can take value 1.

State C: Two successive digits where the first has value 1 and the second value of Ø (1,Ø). The next digit can be either 1 or 2.

State D: Two successive digits where the first has value Ø and the second value of 2 (Ø 2). The next digit can take value 1.

State A generates state B either through Ø or 1, state B generates state B in addition to state C; state C generates state B and state D; state D generates two states, state C and state A. Referring to fig.3.13, from our discussion the following recursive relationship can be deduced.

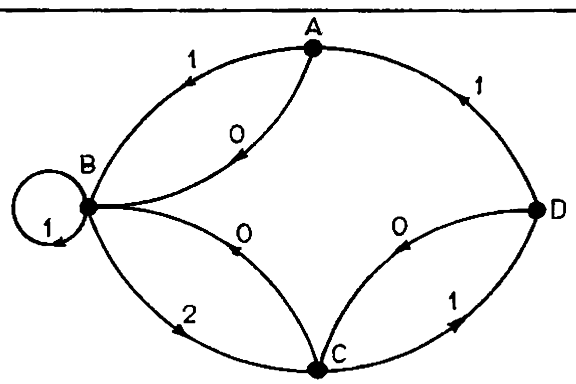


Fig.3.13

$$A(i+1) = D(i)$$

$$B(i+1) = B(i)+A(i)+C(i)$$

$$C(i+1) = B(i) + D(i)$$

$$D(i+1) = C(i) \qquad \ldots\ldots (3.16)$$

Now referring to tree structure in fig. (3.3) and selecting length i, it is possible to get the total number of states. The number of root combination to any length of sequence can be calculated using equation (3.16). Combining all the states mentioned in this equation the number of roots at any length can be written as

$$R(i) = A(i) + B(i) + C(i) + D(i) \text{ and}$$

$$R(i+1) = A(i+1) + B(i+1) + C(i+1) + D(i+1) \qquad (3.17)$$

Replacing equation (3.16) in (3.17) the number of root paths can be found as

$$R(i+1) = A(i)+2B(i)+2C(i)+2D(i) \qquad \ldots\ldots\ldots \qquad (3.18)$$

With appropriate initial conditions A=1, B=C=0 and D=1, the number of roots for general signal and K=1 can be found using the expression (3.18). A few of these are listed in Table III.2

Table III.2, Window size = 3

Column (i)	A(i+1)	B(i+1)	C(i+1)	D(i+1)	No. of Root classes	Roots for binary sequence	Roots for 4 level seqn.
1	1	0	0	1	2	2	4
2	1	1	1	0	3	4	16
3	0	3	1	1	5	6	36
4	1	4	4	1	10	10	94
5	1	9	5	4	19	16	236
6	4	15	14	5	38	26	602
7	5	33	20	14	72	42	1528
8	14	58	47	20	139	68	3882

For comparison the values obtained by Arce [20] for binary signals and Fitch [25] for 4 level quantised signal are also shown. The number of root paths for binary signal is the minimum for all sequence lengths. On the other hand multilevel sequence of the same length have the maximum number of root paths. The root paths using the column sum is the minimum for any sequence of arbitrary length and levels. This is because the multilevel sequence including its trend etc are represented in column sum by its MF window level. This is the major advantage for drawing the tree structure.

3 7 **Summary and conclusion:**

Median filters, being non-linear are not amenable to the elegant approaches of linear filters - convolution and transform analysis. With the input-output characterisation by the wieghting matrix one can construct the output square matrix Y_n directly for certain class of signals. The maximum number of rotations in any column of square matrix is (2K+1). However, this matrix is of limited use in further analysis. On the other hand the median matrix M lends itself for further analysis. The column sum parameter of M indeed characterises the filter by indicating the signal samples that appear at the output and the number of times this happens. For a given window size the limits on column sum, their repetitions and the number of possible combinations of the column sum are all signal dependent and characterise the signal. Periodicity and spikes are reflected in the column sum. Further the column sum and their combinations lend themselves to a tree structure similar to that given by Arce [2Ø]. The root paths and the number of roots generated from this tree structure are more accurate than those of [2Ø] and Fitch [25].

Chapter - IV

INTERPOLATION

Median filtering realisation algorithms are available in both hardware [10,16] and software [14]. In both the methods, sorting to rank the input sequence takes most of the time. In running median operation maximum time is taken for sorting and selection operation for each position of the window. As already discussed in Chapter II, any sequence will become a root sequence under iterated median operation. The maximum number of such iteration is 1/2 (L-2) for sequence of length L. It has been proved in Chapter III that when the input sequence trend and structure [17] are known, the exact number of passes to get a root sequence R is given by $R = \left\lceil \frac{T - (P-1)}{2K} \right\rceil$. In general, atleast a few iterated operations on an input sequence may be required.

To reduce the running median computation time Tukey [6] suggested a method of determining an approximation to a median. Here the data string is arranged in blocks of 3 and the 3 point medians of the blocks are initially determined. An approximation to the 9 point median, the "Ninther", is obtained by considering three 3-point medians. For example the data string (3,1,3,2, 0,1,4,2,7) yields a 3 point median string of (3,1,4) and a ninther (Median of median) is 3 which is only an approximation to the median of the 9 elements. The exact median of the sequence is 2. It can be easily seen that if the data string is monotonically increasing or decreasing then the ninther will yield the exact median. If this method is implemented, the computation time and hardware requirement are reduced. The time delay can be further reduced by employing parallel processing. The present work to be described in this chapter is based on Tukey's idea.

4.1 Interpolated median filter.

It has been discussed in Chapter II that the number of comparators required to implement MF in hardware increases as the window size increases. This increase is nearly exponential as K increases. Even for a 3 point median filter,as is often used in picture processing,at least three comparators are required with a system delay of three samples. It was pointed out by Tukey [6] that an approximation to the true running median may serve an useful purpose. He proposed the use of median of a median by taking W^2 sample sequence with a window size W.

The 'median of a median' method needs atleast (L+1) times filtering for every L^2 length sequence block. On the other hand, in the method suggested here, the median is picked for each window. This median sample replaces the entire window sequence. The window is moved to the next block of new data for median computation, with the median sample replacing each block of window sequences. Thus the output is made of a sequence of neighborhoods. Such a sequence does not undergo any changes in further filtering. The output so obtained is called Fast Convergence Median Filter (FCMF). The output obtained in FCMF is a root sequence and needs only one pass which results in a box car approximation to the original sequence. This is the fastest converging sequence with minimum computation time both in hardware or software methods of realisation. An example to illustrate the smoothing action of ninther, Fast convergence median and running median is given in fig. 4.1.

6 2 5 3 Ø 4 1 3 2
(a) input sequence

3 3 3 3 3 3 3 3 3
(b) Ninther output

5 5 5 3 3 3 2 2 2
(c) Fast convergence median (3 point window)

5 5 3 3 3 1 3 2 2
(d) Running median (3 point window)

Fig. 4.1

The output of ninther is a simple DC term. The FCMF smooths out certain finer variations of the signal which are not filtered out by running median. The filtered output waveform is shown in fig. 4.2.

A better approximation than the ninther and FCMF is possible. It has been noticed that the FCMF introduces a neighborhood equal to the window block. A sharp discontinuity can exist between neighborhoods thus introducing quantisation noise. Further the variance of the median is 57% larger than that of the mean (linear filter) for Gaussian white noise (as discussed in Chapter II). A better approximation is to introduce a monotonic region (linear) between the median window segments. Similar to neighborhood, a monotonic region is also a root and it will not alter the segment median in any way. This is called a Interpolated Median Filter (IMF) and is implemented as follows:

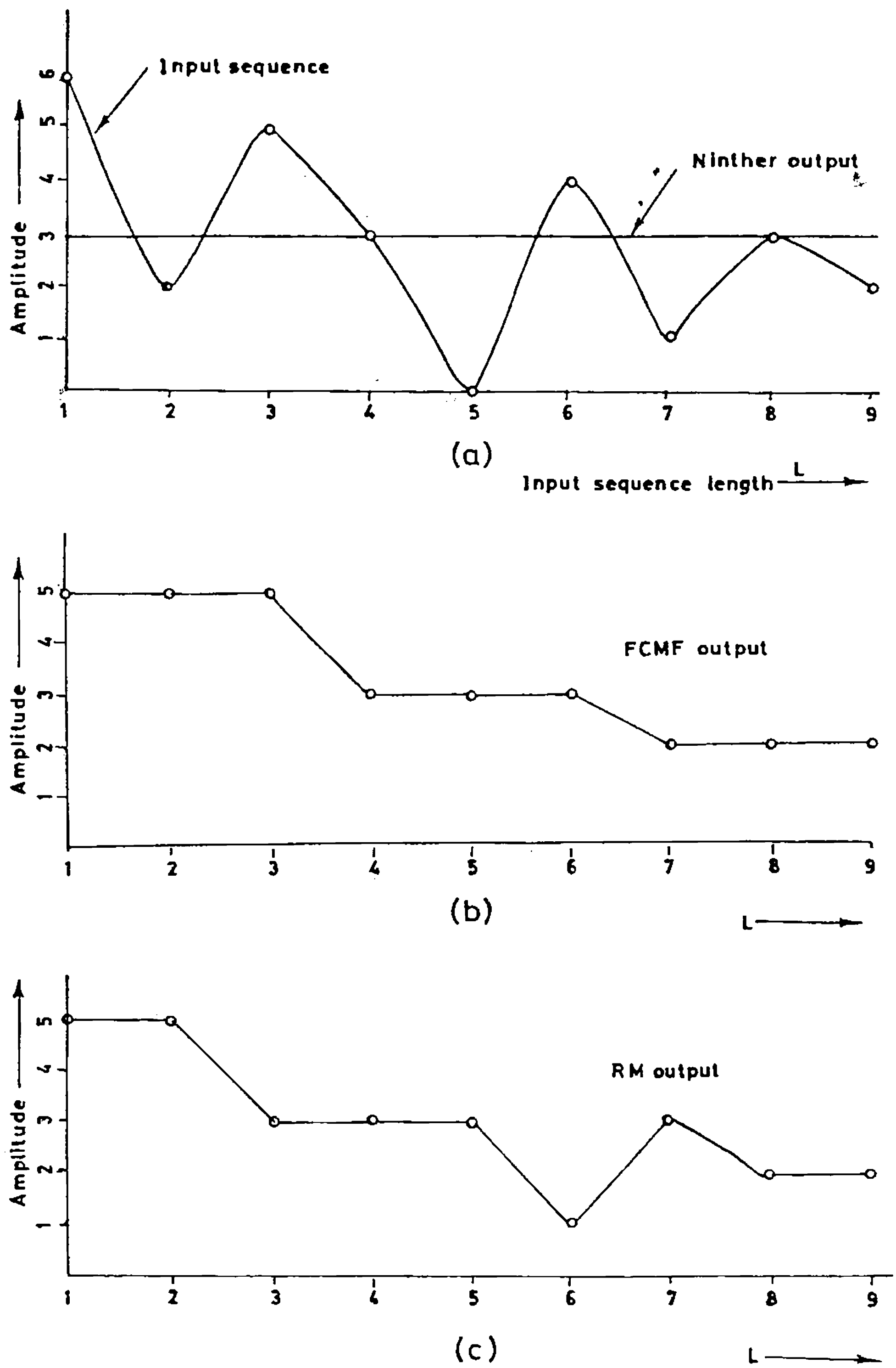
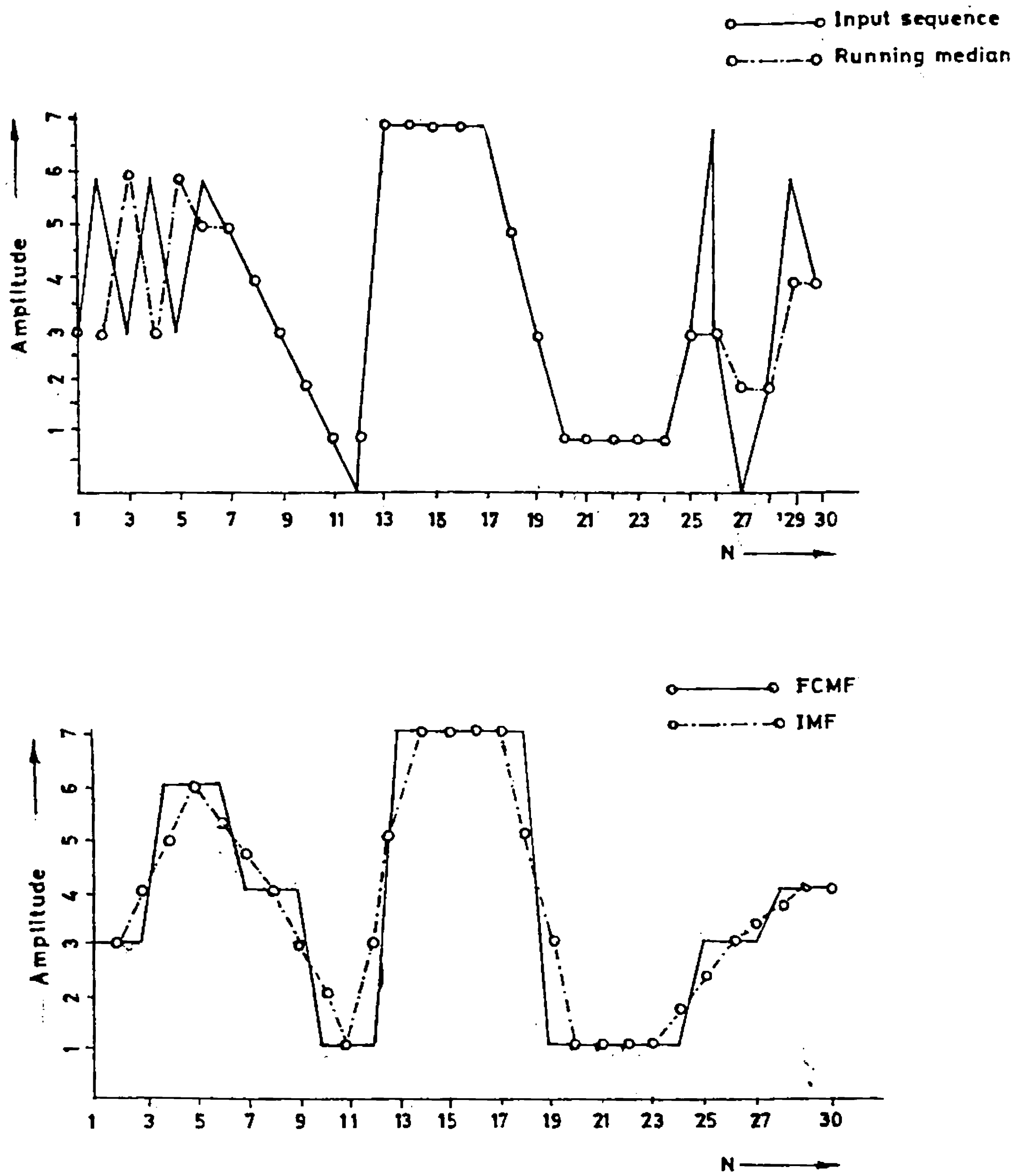

Input sequence
Ninther output
Amplitude
1
2
3
4
5
6
7
8
9
(a)
Input sequence length L
FCMF output
(b)
L
RM output
(c)
L

Fig. 4·2

The median of each segment corresponding to a window size (2K+1) is evaluated (J_i). In addition, the difference between successive segment medians is also evaluated. Let this difference be d_i. The median samples (J_i) replace its center element of the specific window segment. The samples in between two successive segments of medians are now evaluated using d_i. There are 2K samples to be generated in between two known segment medians. These samples are linearly interpolated between the values J_i and $J_{(i+1)}$. A step size of $\frac{d_i}{(2K+1)}$ is added or subtracted from the median segment value J_i to obtain the next sample. Thus the region between true median samples are filled by linear interpolation. This procedure is for all segments to obtain an output sequence. The median output obtained like this with linear interpolation is called Interpolated Median Filter (IMF).To illustrate the IMF an example is shown in fig.4.3. It is clear that as against the box car approximation of FCMF, we now have a linear approximation to the true median output. Possibly other type of interpolation are feasible but to limit the amount of computation only a linear interpolation is attempted in this work.

The comparison between output waveforms of FCMF and IMF is shown in fig. 4.3. The input signal is a random sequence and consists of signal regions which are roots as also segments which



Input sequence
Running median
Amplitude
N
FCMF
IMF
Amplitude
N

Fig 4·3

are not roots. The closeness of the output sequence of IMF to the true RM as well as the FCMF is evident. However, it can also be noticed that for portions of the signal which show oscillatory (bivalued) tendencies, there is very little to choose among the median filters mentioned. This is understandable since the number of passes for such segments to become roots depend on the number of oscillations [17] and approximating these bivalued sequences by a root made up of either neighborhood or monotonic is equally incorrect. A qualitative picture of the performance of all these filter emerges only if the mean square errors (MSE) are compared as is done in the next section.

4.2 Performance Evaluation:

The performance of the FCMF and IMF is evaluated by comparing the mean square error (MSE) of these two filters with that of the running median. For the purpose of a evaluation a 3-point window is considered. A 3-point running median of a signal sequence of length L serves as the reference. The mean square error between (1) the RM and the FCMF output sequence and (2) the RM and the IMF output sequence are computed. The signal considered for this purpose is general though to study the effect of the operations in more detail it is made up of regions clearly representing roots, oscillations etc. (fig. 4.3)

The MSE is calculated and presented in Table IV.A. It can be noticed that the MSE of the IMF is always better than that of the fast convergence median filter. In other words the IMF is a better approximation to the running median than the FCMF. Due to the basic nonlinear nature an analytical evaluation of the MSE or even its bounds is not possible. For a composite signal with roots, monotonic and neighborhood regions with oscillations etc. it can be shown that the MSE of IMF is always lower than that of the FCMF. As a further comparison hardware implementation of Tukey's ninther, FCMF and IMF are considered.

Table IV.A
Window Size: 3

No	Mean Square Error	
	FCMF	IMF
1	1.066	0.947
2	2.166	1.706
3	1.166	0.720

4.3 Implementation

In this section hardware implementations of FCMF and IMF are compared with that of Running Median. The discussion is restricted to window size 3. The arithmatic operations involved in these are defined first and then compared with the running median.

A sequence length L is segmented to window size 3 yielding L/3 segments. Median operation is performed on these segments. These operations involve comparisons which in turn introduce delays for each segment. The median obtained for each segment replaces the entire segment samples as a block. Thus the output consists of L/3 segments of neighborhoods of length equal to the window block. Thus the output obtained in this manner does not undergo any change in repeated passing of MF.

For IMF, the sequence length L is divided into L/(2K+1) segments as in the case of FCMF. The median of each window segment is extracted by comparison method. The median sample replaces the central elements of each window segment. The difference between successive segment median sample is computed (di). The missing 2K samples between the i^{th} and $(i+1)^{th}$ segment median samples are to be approximated. These samples are linearly interpolated with a step size of di/(2K+1). This step size is added or

Table IV.B

S No.	Operation	Running Median	FCMF	IMF
1	Comparison	(L-2K)XNo. of comparators in window (2K+1)	L/(2K+1)x No. of comparators in a window (2K+1)	L/(2K+1)x No. of comparators in a window (2K+1)
2	Arithmetic	Nil	Nil	One division and 3 addition per segment
3	Delays	(L-2)xNo. of delays in window(2K+1)	L/(2K+1)x No.of delays in window (2K+1)	L/(2K+1)x No. of delays in window (2K+1)+ delay due to Arithmetic operation

subtracted successively with the segment median sample to fill the 2K samples. The arithmetic operations involved are addition and division. For window size 3, it is required to interpolate 2 samples which need one division and three addition/subtractions.

The number of operations and savings in hardware and computation time with respect to those of the running median is given in Table IV.B. There is considerable saving of computation time for both FCMF and IMF methods The computation time saving for both FCMF and IMF is (2K+1) times that of the running median. Since this saving is proportional to window size, for large windows the saving is appreciable.

4.4 Image processing

Application of median filtering to picture and image processing is discussed in [3 4 10 16 23].These filters can be either two dimensional filters or seperable median filters as discussed in chapter II. Narendra has shown that both filters perform alike and a separable median filter is more efficient in terms of realisation. As it has been noticed in the preceeding section, FCMF introduces neighborhoods of length equal to the window block while the IMF introduces a monotonically increasing or decreasing region. The latter is similar to a linear smoother between each window block. In this section separable filter image processing is introduced for FCMF/IMF and their performances are compared. The performances of a Separable Fast Convergence Median Filter (SFCMF) and a Separable Interpolated Median Filter (SIMF) are assessed by evaluating the MSE with respect to a Separable Median Filter (SMF) In this context, certain examples are considered which include different types of signal segments like roots, oscillations random parts etc shown in fig. 4.3. Such a signal is passed through the FCMF as well as IMF and the MSE is evaluated with the true RM output as the reference. The results are listed in Table IV.A. The MSE of IMF is always less than the FCMF because of its better approximation to median filter. The SFCMF and SIMF implementation to picture/image processing is now described.

Let the image consist of MxN pixels corrupted by noise. In processing this image a sliding two dimensional window covering odd number of picture elements is passed across the picture from left to right and at every instant the central pixel is replaced by the median of the pixels in the window. The output of this filter depends on the window shape. Narendra has considered processing images by a separable filter using one dimensional windows. In separable filter,first the rows of the picture matrix are processed by one dimensional window and then the columns by the same window.

The separable filter implementation algorithm of FCMF and IMF for image processing is as follows .A line segment window is applied on each row of the image. The row elements are segmented to the size of the window. The median element replaces the entire window segment at the output. The output so obtained gives a new set of MxN output elements. Now, the same line segment window is applied on column segments of the image. The output so obtained is called separable Fast Convergence Filter. The difference between the SMF and SFCMF is that in the former the window slides on the pixels while in the latter a complete window segment is filled with new picture elements. Thus it is possible to reduce the processing time by two thirds for a 3 point window. At the end of each row/column processing if any odd number of elements are left, then these elements can be processed by a smaller window size if possible or the same elements may be filled in the output as such.

The IMF implemented with one dimensional window for processing images is called SIMF. The performance of the SFCMF and SIMF is studied by applying this technique to a picture with and without noise. An image of 32x32 pixels shown in fig. 4.4.a is used for performance evaluation. The output of SMF, SFCMF, SIMF and Moving Average are shown in fig. 4.4 (c) -- (f). The picture is free from noise and it has been found that the first three filters (SMF, SFCMF and SIMF) are preserving the sharp discontinuities of the picture. These filter behaviour is studied on the same image corrupted by (i) white noise (ii) Gaussian noise. In all cases only a window of 3 is considered. The MSE computation results are shown in the Table IV.C. It can be noticed that SIMF is consistantly better in MSE sense than the other for both types of noises.

Table IV. C Window Size: 3

S No.	Filter	Mean Square Error	
		White noise	Gaussian Noise
1	FCMF	666.8037	522.4170
2	SIMF	470.6123	515.8125
3	MA	521.1719	597.3809

The processed image is shown in Fig.4.5 and Fig. 4.6 for white noise and Gaussian noise. The SIMF picture quality is

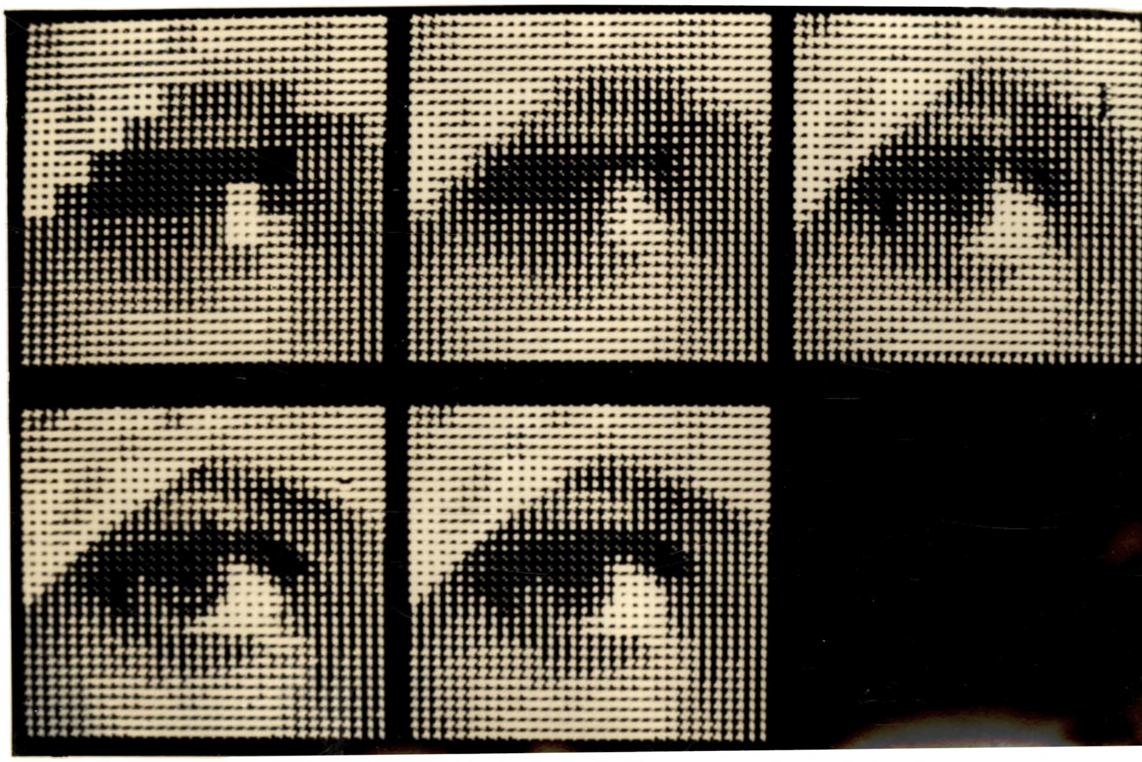


Fig.4.4 (From L to R bottom) (a) Image (b) Seperable MF.

(From L to R top) (c) FCMF (d) IMF (e) Moving Average.

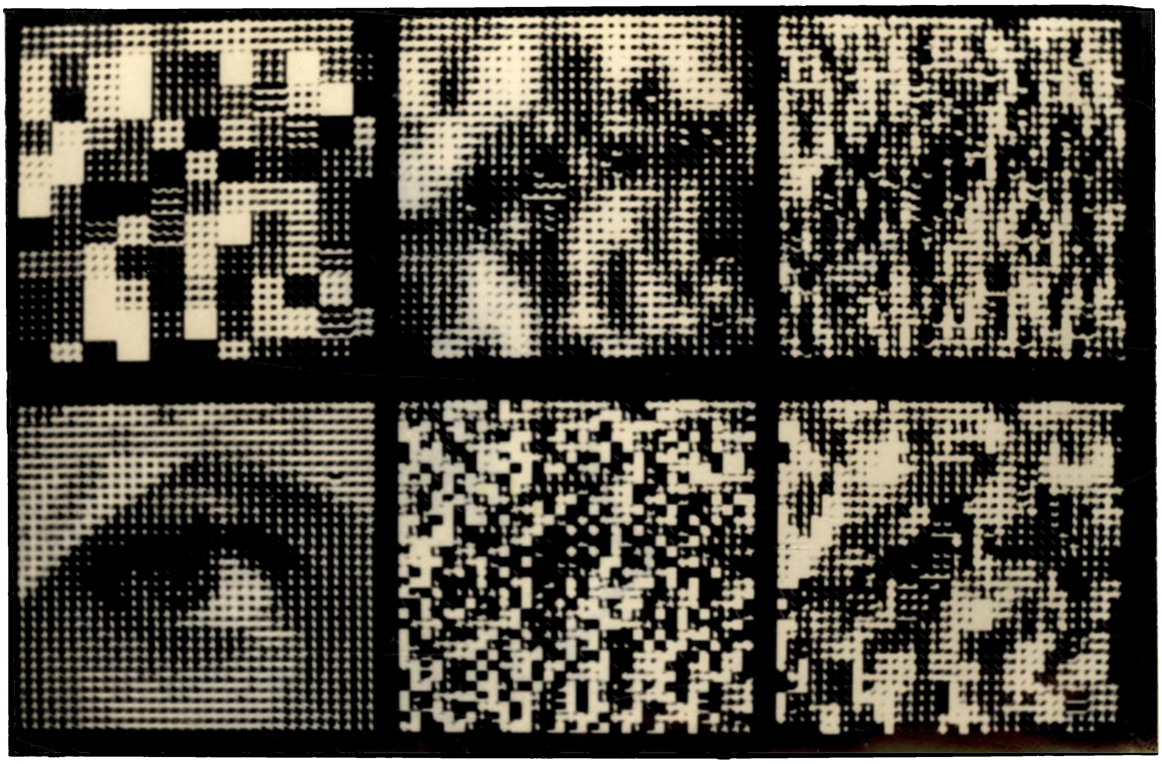


Fig 4.5 (From L to R) (a) Image (b) Image with white noise (c) SMF (From L to R top) (d) FCMF (e) IMF (f) Moving Average

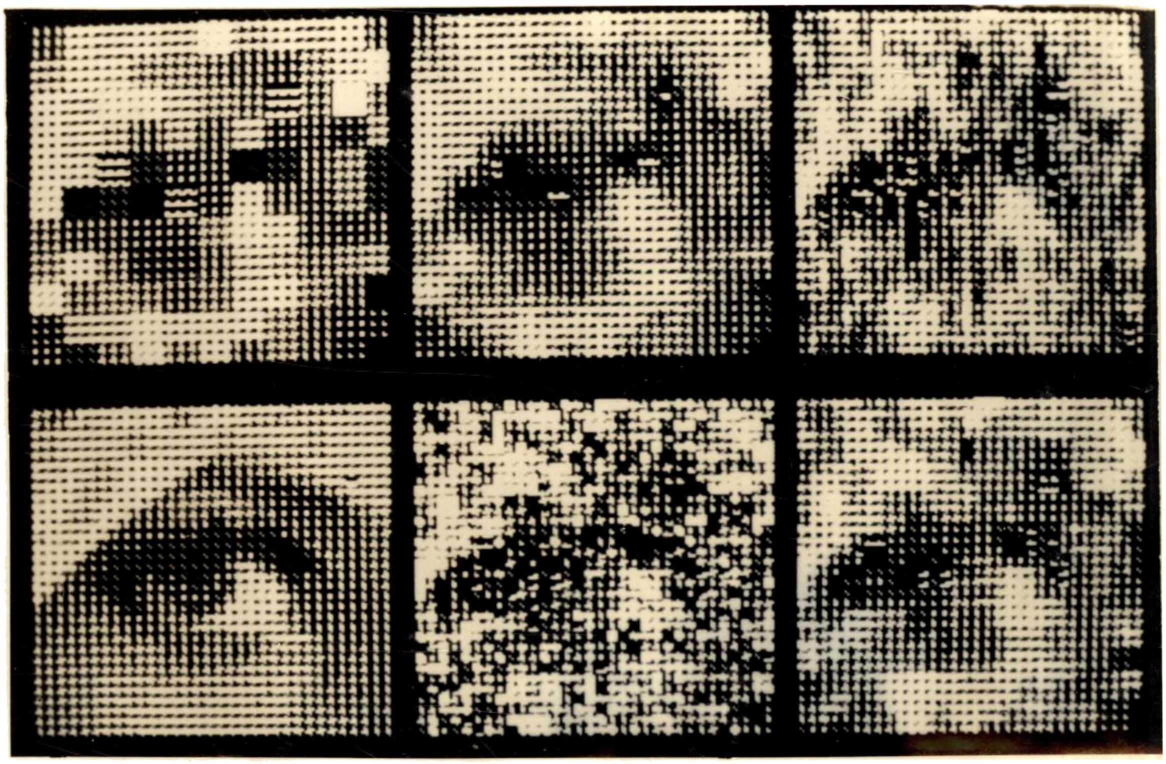


Fig 4.15 (From L to R bottom) (a) Image (b) Image with gaussian noise (c) SMF From L to R top) (d) FCMF (e) IMF (f) (f) Moving Average

identical to that of SMF. Further SIMF preserves constant intensity regions well while retaining the edges. This is not true for SFCMF, SMF and MA. Another feature of SIMF is that at the corners and sharp edges its behaviour is different from both 2D and separable filters. The SMF converts a region of fluctuating grey levels to two smears of black while the SIMF preserves more number of grey levels in the same region. In other words the SIMF improves contrast in the picture. Thus the SIMF is a better approximation to the available 2D or separable filter techniques.

4.5 Conclusion:

FCMF and IMF filtering techniques have distinct advantage in picture and image processing. These two methods are simple to realise both in terms of time and hardware. The processed picture quality is comparable in terms of MSE and contrast. This is evident from fig. 4.5 and fig. 4.6 for white and Gaussian noises, respectively.

Chapter V

FREQUENCY DOMAIN ANALYSIS

Though median filters have been applied for speech and picture processing, it has not been possible to define their characteristics in frequency domain because of their non-linear nature. However attempts have been made to categorise their behaviour. Justusson [9,18] considered a harmonic signal and evaluated the variance which is expanded as a Fourier Series. He established that the median filtered version of this signal has the same covariance as that of a continuous time process. Further, he proved that the spectral response of median filters is the same as that of moving averages. Vellman [15] has conducted extensive simulation studies and his results are quite similar to those of Justusson. The difference between the analysis of Justusson and Vellman is that Vellman treats MF as one of the nonlinearities he considered, while Justusson treats only MF. Vellman finds the power transferred by the MF from fundamental to its harmonies and presents the result in terms of sidelobe levels. By concatanating two different order MFs Tyan [8,18] has shown that the sidelobe level can be further reduced.

$$Y_n = \phi_2 (\phi_4 (x_n)) \qquad \ldots\ldots \qquad (5.1)$$

where ϕ_4 and ϕ_2 are even numbered 4 and 2 point running medians, respectively, has lower sidelobe levels.

The frequency domain characteristics in these works do not truly represent the frequency response in the conventional sense.

For example fig. 5.1 shows Justusson's simulation result rather than an analysis as he considers only a harmonic signal. The same is the case with the results of Vellman and Tyan. It is also obvious that it is not possible to define a 'Frequency response function' for a median filter similar to that of a linear filter. However, alternate characterisation is possible by observing these. The output of a median filter is basically a subset of the input sequence and hence DFT of the output sequence can be related to that of the input sequence. In this chapter after introducing step response, impulse response etc as done by Justusson, analysis proceeds to present methods of determining the relationship between the input and output DFTs. It is shown that in a few cases it is possible to obtain DFT of the output sequence by simple arithmetic.

5.1 IMPULSE, STEP RESPONSES

Digital filters like most linear systems are described in terms of their impulse, step or frequency response and these are all interrelated.

Median filtering eliminates impulses for all K. Hence it can be seen that impulse response of a median filter is zero.

Let $y_{(n)} = \phi_K (x_n)$ where ϕ_K is the running median for a given K. This can be written as mentioned in Chapter III in

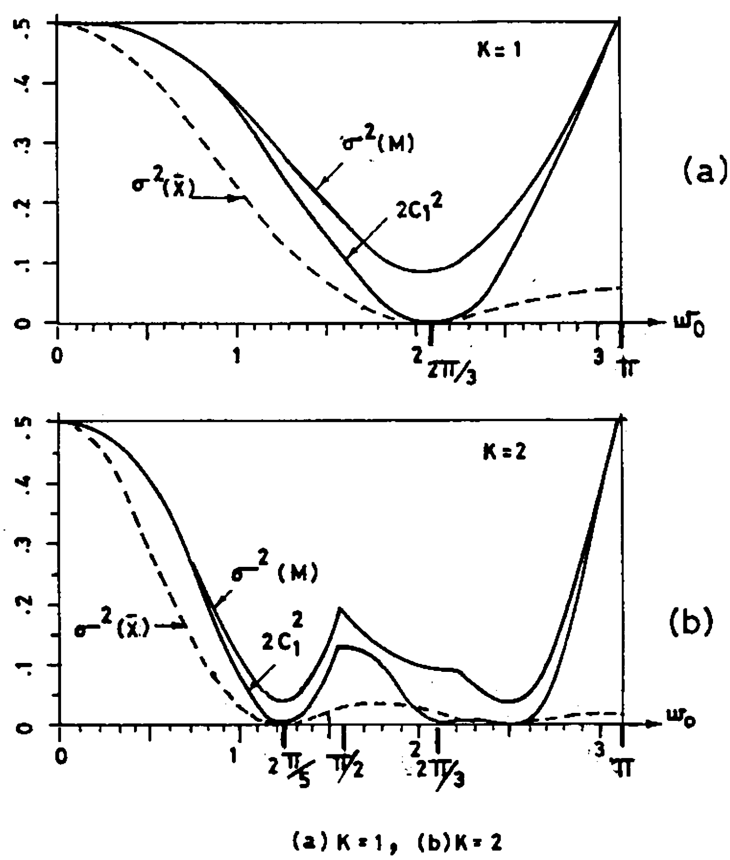


Fig 5·1 FILTERING OF A COSINE WAVE

terms of weighting matrix for K=1.

i.e. $$y_{(n)} = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix} \cdot \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

or $$y_{(n)} = \sum_{i=1}^{3} a_i \, x_i \qquad \ldots\ldots (5.2)$$

when the input is $\delta(n)$

$Y(n) = h(n) = \emptyset$

To define the step response of a MF, let $x_n = 1$ for all n .
Output $y_{(n)} = \phi_K(x_n) = 1$ for all n because x_n is a neighborhood for all K. Therefore the step response of MF for any K is UNITY.

5.2 FREQUENCY RESPONSE:

Let x_n be a sequence of length L. On passing through a median filter of window (2K+1), the median filter removes the spiky noise less than K samples wide and preserves other sharp edges. Both the spiky noise and sharp edges are high frequency components. The effect of MF on deterministic signal for different period is analysed in time domain in Chapter III. The subsequent discussion is to characterise the median filtering effect in frequency domain.

Let x(n) be the input sequence passed through a (2K+1) window Median Filter and the output sequence be $x_m(n)$. Now x(n) can be written as

$$[x(n)] = [x_m(n)] + [x_r(n)] \qquad \ldots\ldots \qquad (5.3)$$

where all the vectors are of length L. Here the vector $x_r(n)$ is obtained from $[x(n)]$ amd $[x_m(n)]$. The output vector $x_m(n)$ is interpreted as the smooth part of the signal x(n) made up of samples which pass through the MF without any change or at worst replaced by another sample of x(n), if any trend change over is present, in the same window and $[x_r(n)]$ is interpreted as the rough part of the signal. The rough part of the vector takes zero values where the input signal is smooth and non zero at the trend change over points and impulses.

Example: Let x(n) be the signal vector and its trend be changing every third sample.

The input $[x(n)] = x_0 \; x_1 \; x_2 \; x_3 \; x_4 \; x_5 \; x_6 \; x_7 \; x_8 \; x_9 \ldots x_{(n)}$

MF Output $[x_m(n)] = \square \; x_1 \; x_2 \; \square \; x_4 \; x_5 \; \square \; x_7 \; x_8 \; \square \ldots x_{(n)}$

$$[x_r(n)] = [x(n)] - [x_m(n)]$$

where $\square$ indicating the trend change over is being replaced by its neighbouring samples in the same window. It is to be noted that the rough and smooth parts defined here are different from those defined by Rabiner et. al. [2]. In frequency domain the DFT coefficients of $[x(n)]$, $[x_m(n)]$ and $[x_r(n)]$ are computed and compared. By linearity of DFT it can be seen that

$$\text{DFT}\,[x(n)] = \text{DFT}\,[x_m(n)] + \text{DFT}\,[x_r(n)]$$

or

$$X_n(f) = X_m(f) + X_r(f) \qquad \text{or}$$

$$X_m(f) = X_n(f) - X_r(f) \qquad \ldots\ldots \quad (5.4)$$

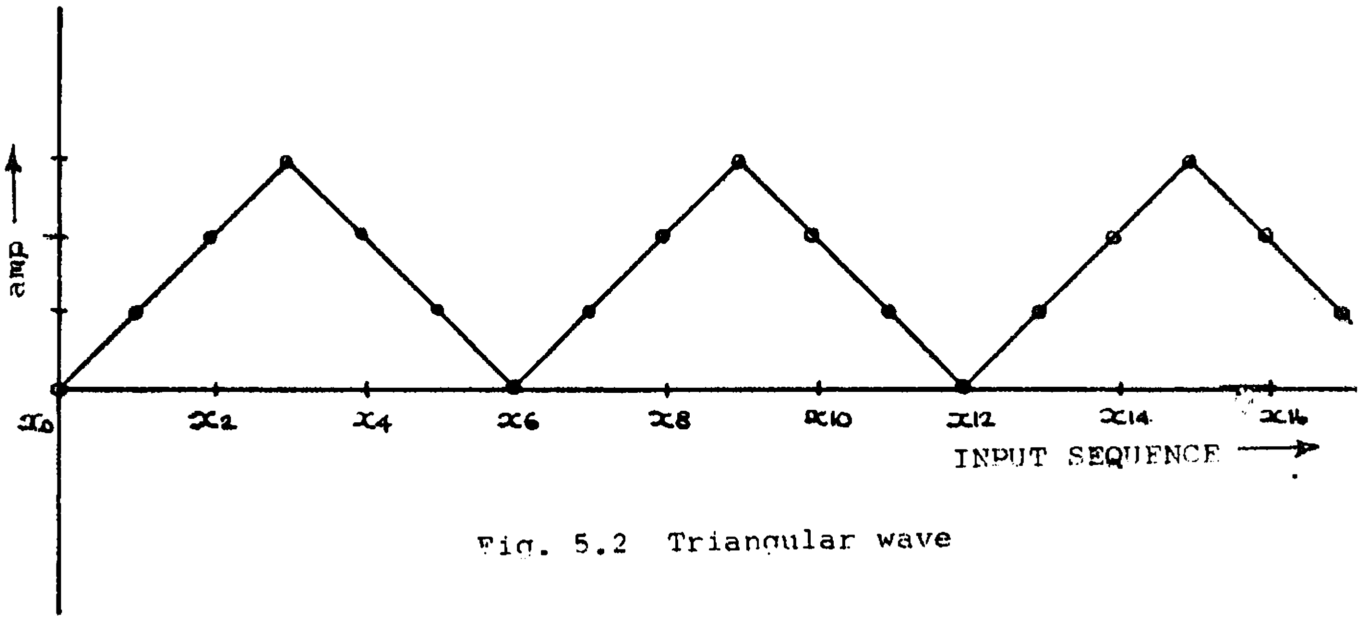


Fig. 5.2 Triangular wave

second brackets respectively. The subscript pi of X is the periodicity at which the trend changes for i = 0,1,2 ...... Further the subscript pi is split into odd and even for the sake of convenience. That is

$$X_0 = C_0 + (X_{pi\ even} + X_{pi\ odd}) \quad .... \quad (5.7)$$

$$= C_0 + (P_0 + Q_0)$$

The corresponding $X_m(f)$ DFT coefficient of median filter output is

$$X_{m0}(f) = C_0 + (X^1_{pi\ even} + X^1_{pi\ odd}). \quad ..... \quad (5.8)$$

For the signal under consideration it can be seen that the maximum and minimum samples are being replaced by preceeding and succeeding samples respectively. Therefore

$$x_{pi\ even} = x_{(p+1)\ i\ even}$$

$$x_{pi\ odd} = x_{(p-1)\ i\ odd}$$

Now, the DFT coefficient for median output can be written as

$$X_{m0}(f) = C_0 + (A_0 + B_0) \quad \text{where}$$

$$A_0 = (x_0 + x_6 + x_{12}) \quad \text{and}$$

$$B_0 = (x_3 + x_9 + x_{15})$$

The DFT coefficient $X_1(f)$ may be written as

$$X_1(f) = C_1 + (x_0 + x_3 e^{-j\frac{2\pi}{N}\cdot 3} + x_6 e^{-j\frac{2\pi}{N}6} + x_9 e^{-j\frac{2\pi}{N}9} + x_{12} e^{-j\frac{2\pi}{N}12} + x_{15} e^{-j\frac{2\pi}{N}\cdot 15}$$

where c_1 is the complex number due to the computation of invariant part of the input signal. The variable part of the samples are grouped in the bracket term. Further this can be split into two portions namely signal minimum and signal maximum as follows

$$X_1(f) = C_1 + (x_0 + x_6 e^{-j\frac{2\pi}{N}\cdot 6} + x_{12} e^{-j\frac{2\pi}{N}\cdot 12} + (x_3 e^{-j\frac{2\pi}{N}\cdot 3} + x_9 e^{-j\frac{2\pi}{N}\cdot 9} + x_{15} e^{-j\frac{2\pi}{N}\cdot 15} \qquad (5.9)$$

$= C_1 + (P_1 + Q_1)$ where P_1 and Q_1 are complex and

$$P_1 = (x_0 + x_6 e^{-j\frac{2\pi}{N} 6} + x_{12} e^{-j\frac{2\pi}{N} 12})$$

$$Q_1 = (x_3 e^{-j\frac{2\pi}{N}\cdot 3} + x_9 e^{-j\frac{2\pi}{N} 9} + x_{15} e^{-j\frac{2\pi}{N} 15})$$

The DFT coefficients of MF output $X_{m1}(f)$ is

$$X_{m1}(f) = C_1 + (A_1 + B_1) \quad .... \quad (5.10)$$

where

$$A_1 = \left(x'_0 + x'_6 e^{-j\frac{2\pi}{N} 6} + x'_{12} e^{-j\frac{2\pi}{N} 12}\right)$$

$$B_1 = (x'_3 e^{-j\frac{2\pi}{N}\cdot 3} + x'_9 e^{-j\frac{2\pi}{N} 9} + x'_{15} e^{-j\frac{2\pi}{N} 15})$$

similarly it is possible to write

$$X_{(N-1)}(f) = C_{(N-1)} + P_{(N-1)} + Q_{(N-1)} \quad .....(5.11)$$

The corresponding coefficient of MF output is

$$X_{m(N-1)}(f) = C_{(N-1)} + A_{(N-1)} + B_{(N-1)} \quad ...(5.12)$$

The coefficient ratio

$$\frac{X_{mk}(f)}{X_k(f)} = \frac{C_k + (A_k + B_k)}{C_k + (P_k + Q_k)} \quad ..... \quad (5.13)$$

So far the discussion is restricted to a specific signal (triangular wave). The same argument of maxima/minima change over points in MF output is applicable to all periodic signals. It is interesting to see the MF output for periodic signals in frequency domain. It is observed that

$$\frac{X_{mk}(f)}{X_k(f)} = \text{a constant ratio}$$

In order to place this in evidence a few specific illustrations are considered.

Case 1:

Let the input sequence x(n) of a triangular wave be

Ø 1 2 1 Ø 1 2 1 Ø 1 2 1 Ø 1 2 1

The corresponding MF output for window 3 (K=1) is

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

The output is a step function (D.C shift). The frequency response is an impulse function. The input output frequency response is shown in fig. 5.3.

Case II(a) :

A symmetric triangular wave x(n) is given by

Ø 1 2 3 2 1 Ø 1 2 3 2 1 Ø 1 2 3

The MF output $x_m(n)$ for Window size 3 (K=1) is

1 1 2 2 2 1 1 1 2 2 2 1 1 1 2 2

The rough part of the input sequence $x_r(n)=x(n) - x_m(n)$

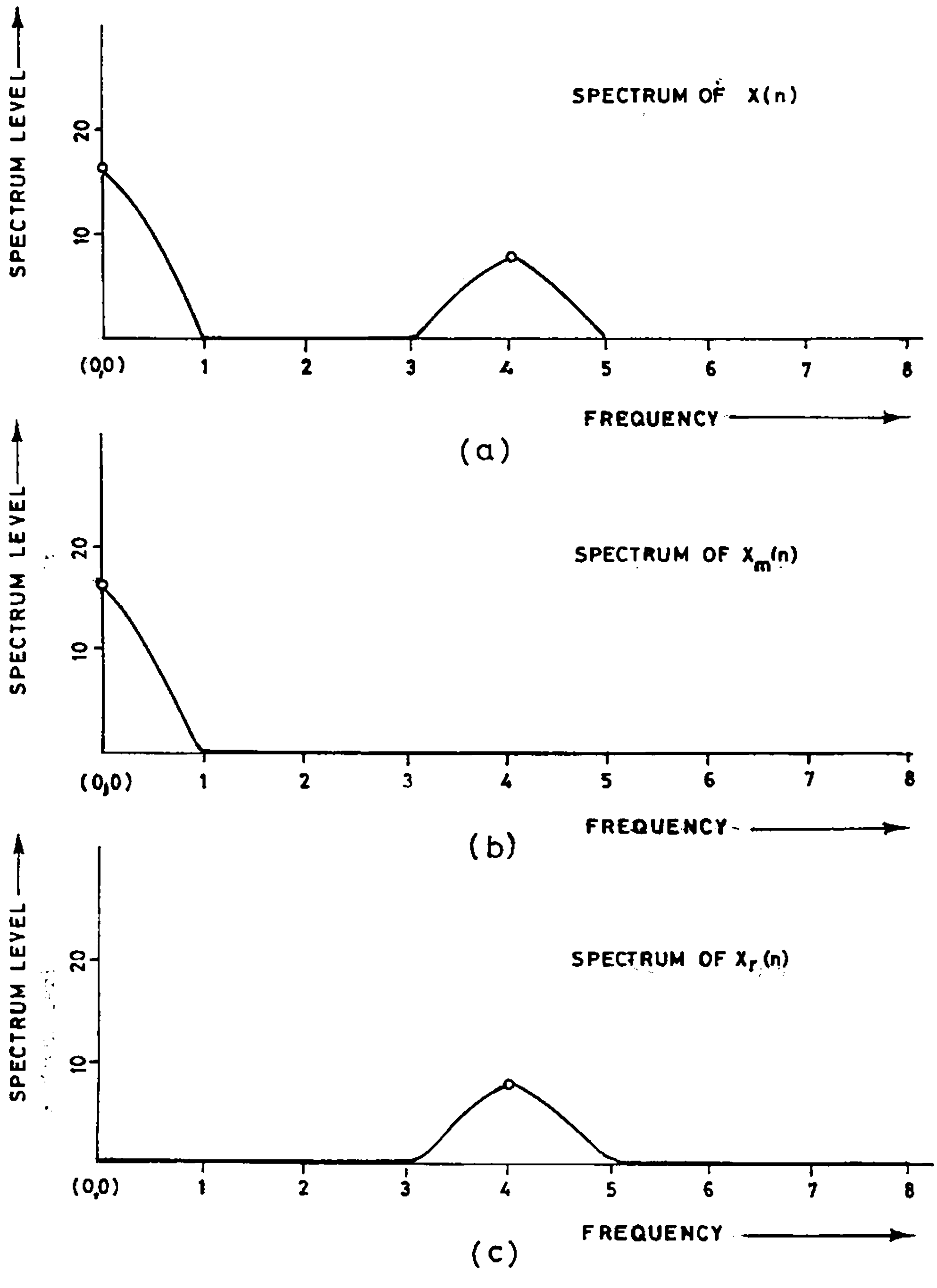

SPECTRUM LEVEL
20
10
SPECTRUM OF X(n)
(0,0)
1
2
3
4
5
6
7
8
FREQUENCY
(a)
SPECTRUM LEVEL
20
10
SPECTRUM OF $X_m(n)$
(0,0)
1
2
3
4
5
6
7
8
FREQUENCY
(b)
SPECTRUM LEVEL
20
10
SPECTRUM OF $X_r(n)$
(0,0)
1
2
3
4
5
6
7
8
FREQUENCY
(c)

Fig 5.3

That is

$$-1\ 0\ 0\ 1 \quad 0\ 0\ -1\ 0 \quad 0\ 1\ 0\ 0 \quad -1\ 0\ 0\ 1$$

The frequency responses of $x(n)$, $x_m(n)$ and $x_r(n)$ are shown in fig.5.4. The DFT coefficient

$$X_0(f) = (x_1 + x_2 + x_4 + x_5 + x_7 + x_8 + x_{10} + x_{11} + x_{13} + x_{14}) +$$

$$(x_0 + x_6 + x_{12}) + (x_3 + x_9 + x_{15}) = 15 + (0+9)$$

$$X_{m0}(f) = 15 + (3+6)$$

The ratio $\dfrac{X_{m0}(f)}{X_0(f)} = 1$

$$X_1(f) = \left[x_1 e^{-j22.5} + x_2 e^{-j45} + x_4 e^{-j90} + x_5 e^{-j112.5} + x_7 e^{-j157.5} + x_8 e^{-j180} + x_{10} e^{-j225} + x_{11} e^{-j147.5} + x_{13} e^{-j292.5} + x_{14} e^{-j315}\right] + \left[(x_0 + x_6 e^{-j135} + x\, e^{-j270}) + (x\, e^{-j675} + x\, e^{-j202.5} + x\, e^{-j337.5})\right]$$

$$= (-0.9686 - j0.4274) + \left[0 + (1.1478 - j0.4752)\right]$$

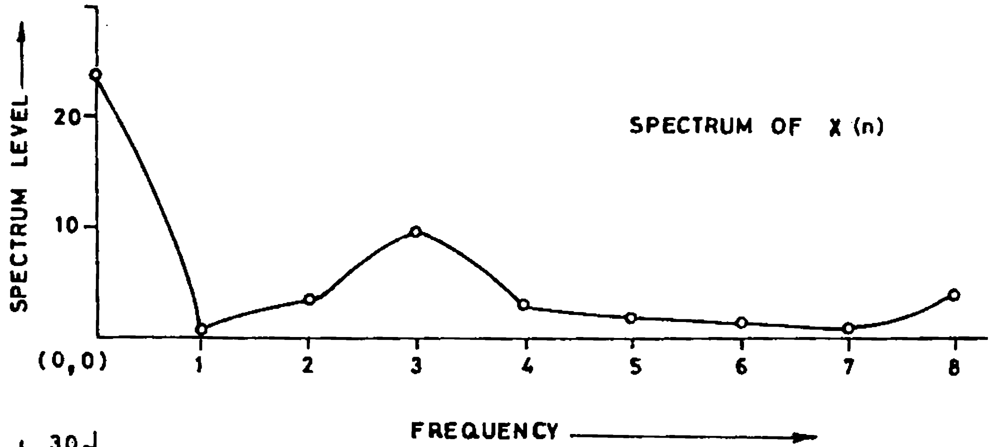
$$X_{m1}(f) = (-0.9686 - j0.4274) + \left[(0.273 + j0.293) + (0.7652 - j0.3172)\right]$$

$$\frac{X_{m1}(f)}{X_1(f)} = 0.5$$

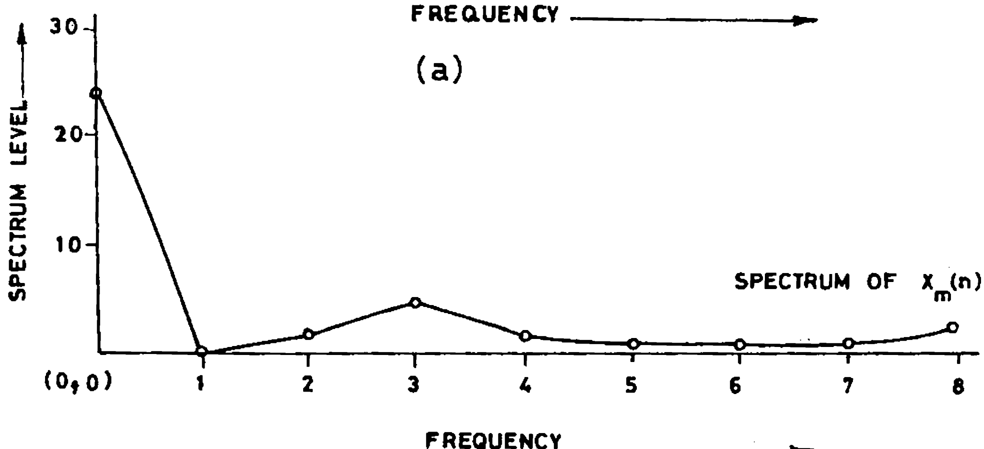
Similar computation of DFT coefficients yield

$$\frac{X_{m2}(f)}{X_2(f)} = \frac{X_{m3}(f)}{X_3(f)} = \cdots \cdots = \frac{X_{m15}(f)}{X_{15}(f)}$$

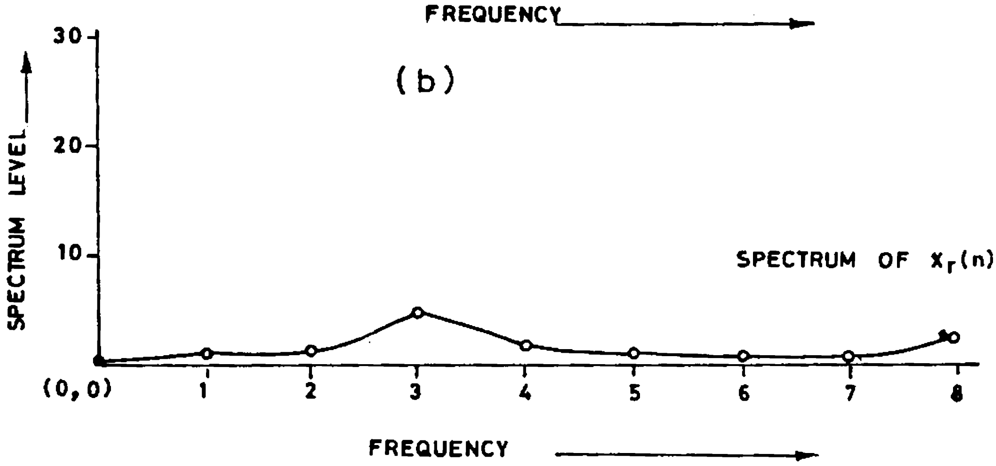
$$= 0.5$$


SPECTRUM LEVEL
20
10
SPECTRUM OF X (n)
(0,0)
1
2
3
4
5
6
7
8
FREQUENCY

(a)


SPECTRUM LEVEL
30
20
10
SPECTRUM OF $X_m(n)$
(0,0)
1
2
3
4
5
6
7
8
FREQUENCY

(b)


SPECTRUM LEVEL
30
20
10
SPECTRUM OF $X_r(n)$
(0,0)
1
2
3
4
5
6
7
8
FREQUENCY

(c)

Fig 5·4

The DFT coefficient ratio is 0.5 except for the D.C. term. The MF has wiped out the maxima and minima samples replacing them by the adjacent samples. This has resulted in a neighborhood in the output. The sum of samples at the input and the MF output are the same. This yields a ratio of $X_{m0}(f)$ to $X'_0(f)$ as unity.

Case II(b):

In this example, the number of samples in a period is maintained the same as in the case II(a) except non-uniform step is introduced. The input sequence is as follows:

0 1 4 5 4 1 0 1 4 5 4 1 0 1 4 5

The MF output of window size 3 (K=1) is

$\boxed{1}$ 1 4 4 4 1 1 1 4 4 4 1 1 1 4 $\boxed{4}$

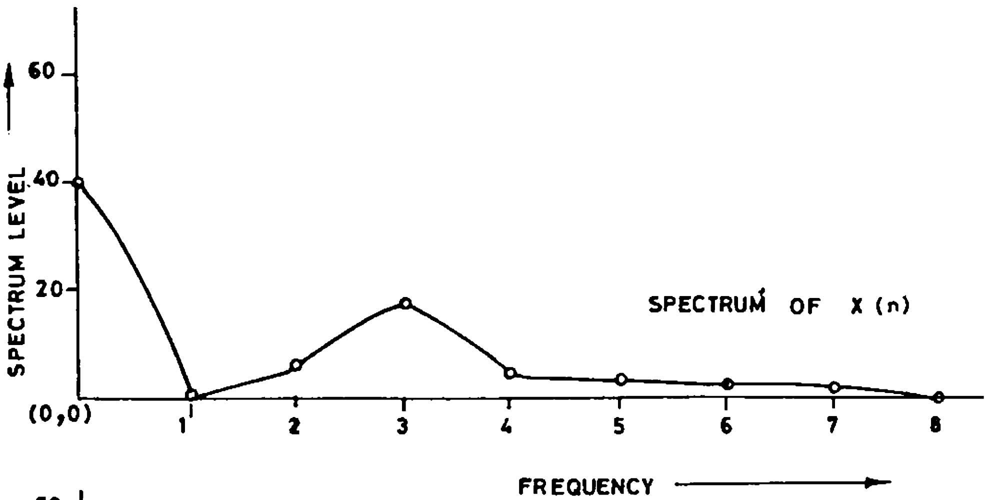
The difference between the input and output of the MF is given by

-1 0 0 1 0 0 -1 0 0 1 0 0 -1 0 0 1

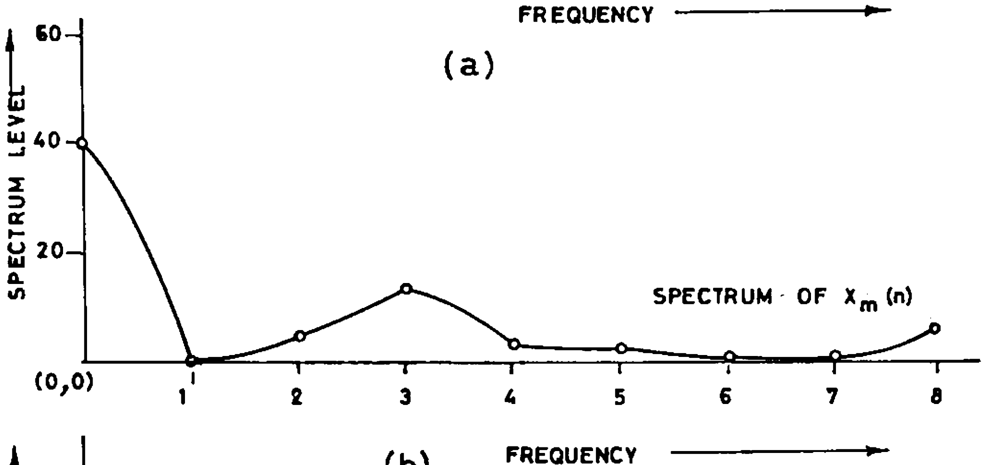
The frequency response plot of X(n), $X_m(n)$ and $X_r(n)$ is given in fig.5.5. It is to be noted that the ratio $X_{m0}(f)$ and $X_0(f)$ is unity and the ratio of the other DFT coefficients are

$$\frac{X_{m1}}{X_1} = \frac{X_{m2}}{X_2} = \ldots\ldots = \frac{X_{m15}}{X_{15}} = 0.75$$

It is to be noted from fig.5.4 and fig.5.5 that the median filter acts as a spectrum subtracting filter. Though the analysis is carried out for periodic signal for simplicity, it is applicable to all class of signals.

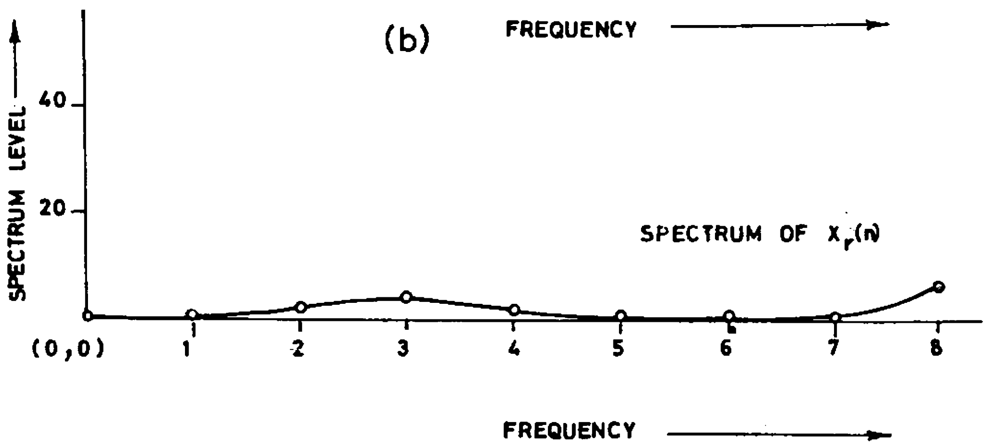

SPECTRUM LEVEL
60
40
20
(0,0)
1
2
3
4
5
6
7
8
SPECTRUM OF X (n)
FREQUENCY

(a)


SPECTRUM LEVEL
60
40
20
(0,0)
1
2
3
4
5
6
7
8
SPECTRUM OF X_m (n)

(b)

FREQUENCY


SPECTRUM LEVEL
40
20
(0,0)
1
2
3
4
5
6
7
8
SPECTRUM OF X_r(n)
FREQUENCY

(c)

Fig 5·5

Case III:

The number of samples in a period is increased for a symmetric triangular wave. The signal sequence MF output sequence and the rough part of median filter are given in fig.5.6.

0	1	2	3	4	5	4	3	2	1	0	1	2	3	4	5

(a)

1	1	2	3	4	4	4	3	2	1	1	1	2	3	4	4

(b)

-1	0	0	0	0	1	0	0	0	0	-1	0	0	0	0	1

(c)

Fig. 5.6 (a) Input Sequence, (b) MF Output of window size 3, (c) the difference of (a) - (b)

It may be noted from fig. 5.6(c) that the sequence has values other than zero only when there is a change in signal trend. The frequency domain plot is given in fig. 5.7. The DFT coefficient ratio of the sequence is

1, 0.8032, 0.8153, 0.8487, 1.00, 0.2789, 0.6132, 0.6572, 0.0

The variation in the ratio is because of change in A_k and B_k values due to MF operation as shown in equation 5.13. Thus the frequency response of a median filter is definable mathematically and easy to visualise for deterministic signals using DFT algorithm.

5.3 Conclusion:

For median filters, it is difficult to define a transfer function, and hence in general frequency

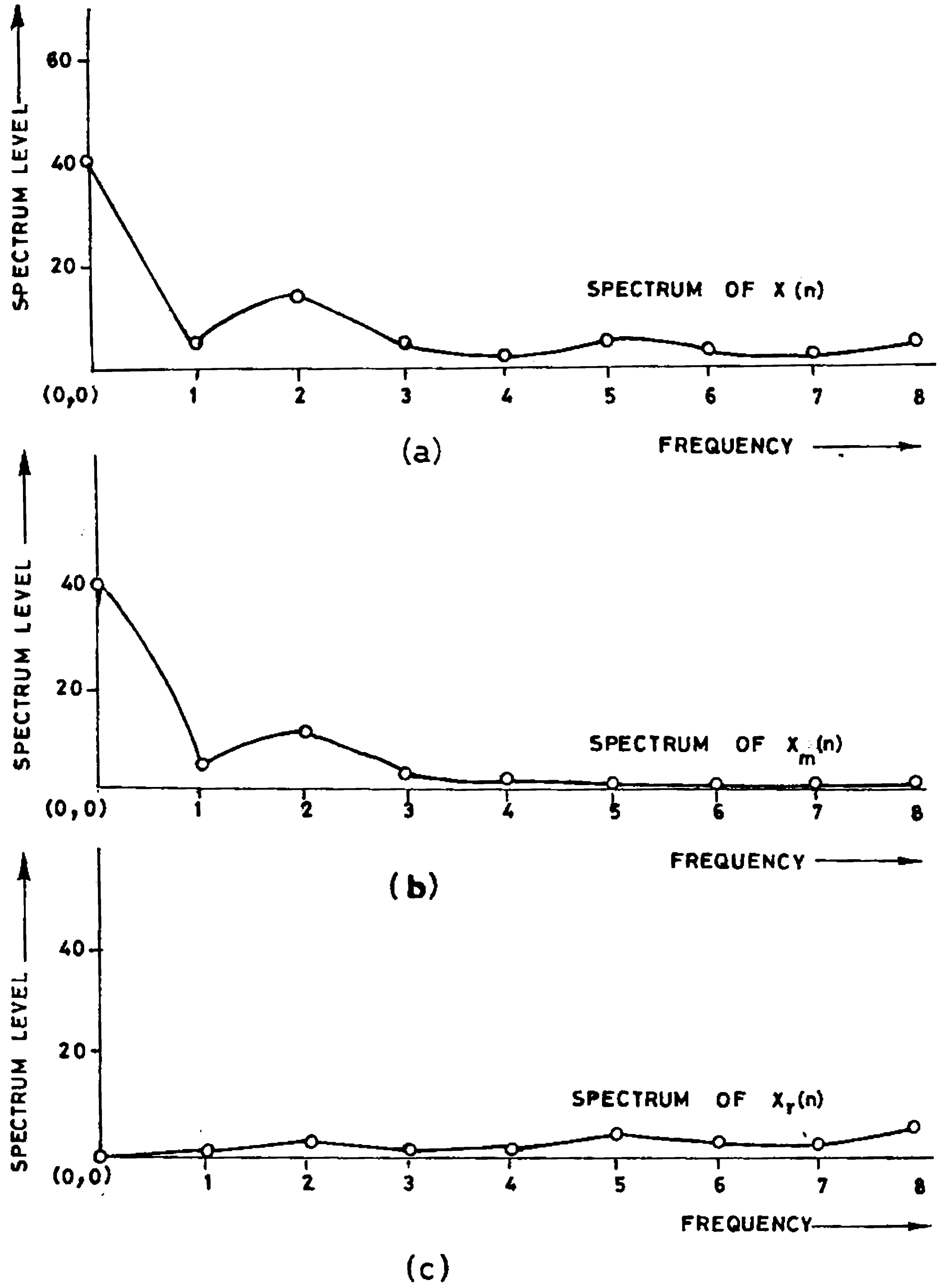

SPECTRUM LEVEL
60
40
20
SPECTRUM OF X (n)
(0,0)
1
2
3
4
5
6
7
8
(a)
FREQUENCY
SPECTRUM LEVEL
40
20
SPECTRUM OF X_m(n)
(0,0)
1
2
3
4
5
6
7
8
FREQUENCY
(b)
SPECTRUM LEVEL
40
20
SPECTRUM OF X_r(n)
(0,0)
1
2
3
4
5
6
7
8
FREQUENCY
(c)

Fig 5.7

domain descriptions are incomplete. It is difficult to categorise them as low pass, high pass etc. However, MF exhibits certain other interesting albeit signal dependent frequency domain properties.

For instance, for all class of signals it operates as a "spectrum subtraction filter". For yet another class of signals the DFT coefficients bear a constant ratio thus leading one to conclude that it can operate as a "spectrum subtractor" and/or"spectrum scaler" depending on the input signal. For random signals, MF exhibits its true nonlinear characteristics and defies categorisation in the sense that one does for linear filters. However, the spectrum ratio output seems to be decomposable into polynomials. It is hence suggested that the frequency domain characteristics of MF is basically one of two classes; (1) spectrum arithmetic operation and (2) spectrum distortion.

Chapter VI

REALISATION AND APPLICATIONS

Median filtering amounts to producing at the output a sample chosen from amongst the input sample x(n), so that smaller and greater values of input occur with equal frequency. Extraction of median requires sorting the samples and hence comparisons are needed. Special algorithms are required to speed up this operation so that online processing can be employed. In Chapter II, some of the available algorithms were discussed. Those methods, in general, are not optimum either in terms of hardware or delay and no clear cut general structures are available for a window (2K+1). Added to these there is neither flexibility for the window size nor for n^{th} ranked operation. In this chapter two new algorithms are presented, one in terms of minimum hardware and the other in terms of minimum delay suitable for VLSI implementation. The latter part of this chapter is devoted to applications of median filtering to underwater target detection as a Ranked CFAR processor, to picture processing for feature extraction and to speech processing for separting the voiced and unvoiced signal and to formant number prediction. The results are discussed in detail.

A. Hardware realisation

6.1 Comparator Method:

Hardware and software algorithms for median have already been discussed in Chapter II. The new algorithm suggested here requires only (2K+2) comparators to find (2K+1) point median. This algorithm has the flexibility of changing the window size which is generally difficult in hardware realisation. Once the median filter is built for a specified window, it has the provision to process data for smaller windows. Besides this n^{th} ranked (non-median) operation is also possible. The proposed hardware structure can efficiently work with no change in hardware and with a nominal alteration in the external interconnections for different window sizes and n^{th} ranked operation.

Detailed hardware realisation for a 5 point (K=2) RM is shown in Fig. 6.1. There are (2K+1) buffers for a given K. Initially the first (2K+1) data of an input sequence x_1 is strobed. The Median Counter M(C) is initialised to zero. All the buffers contents are compared simultaneously with M(C). The comparator outputs,((Buffer)=M(C)) coincidences are brought out as address to a read only memory. The memory output gives the number of buffers which are equal to M(C) at any instant of M(C) clock. The present output of the memory is added with that of the past for every clock period M(C) until $SUM \geq (K+1)$. When the $SUM \geq (K+1)$, the clock M(C) is inhibited. This ensures that the median value of the window sample corresponds to the M(C) count.

The data are inputted to the filter and strobed in latches in modulo (2K+1). In this operation the latest data are overwritten on the oldest data. This algorithm is independant of the input structure. That is,any combinations of neighborhoods, edges, impulses, oscillations etc. The only constraint is that the data update interval time $Td \geq q\ Tc$, where q is the quantisation

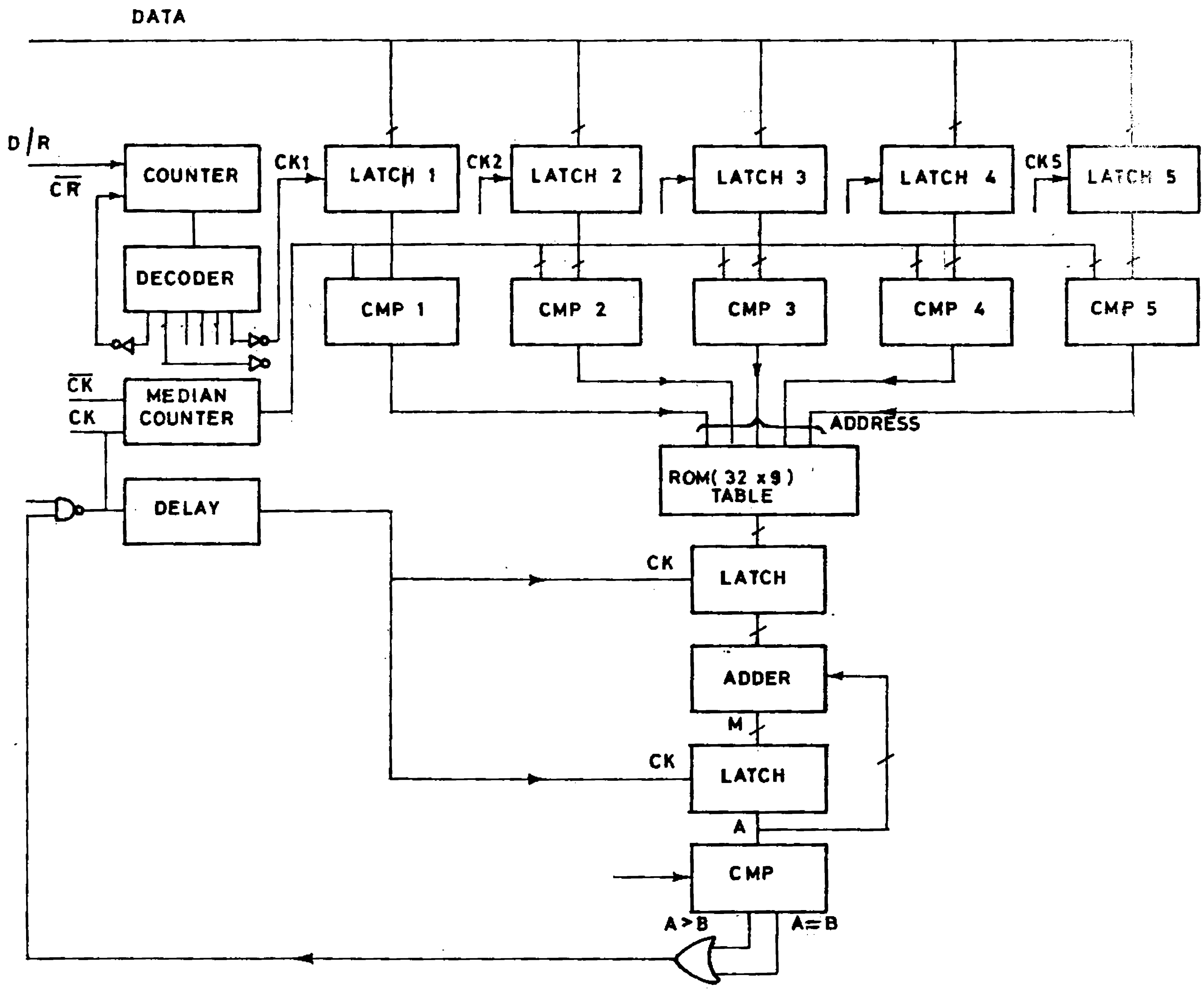


Fig 6·1 FIVE POINT MEDIAN FILTER

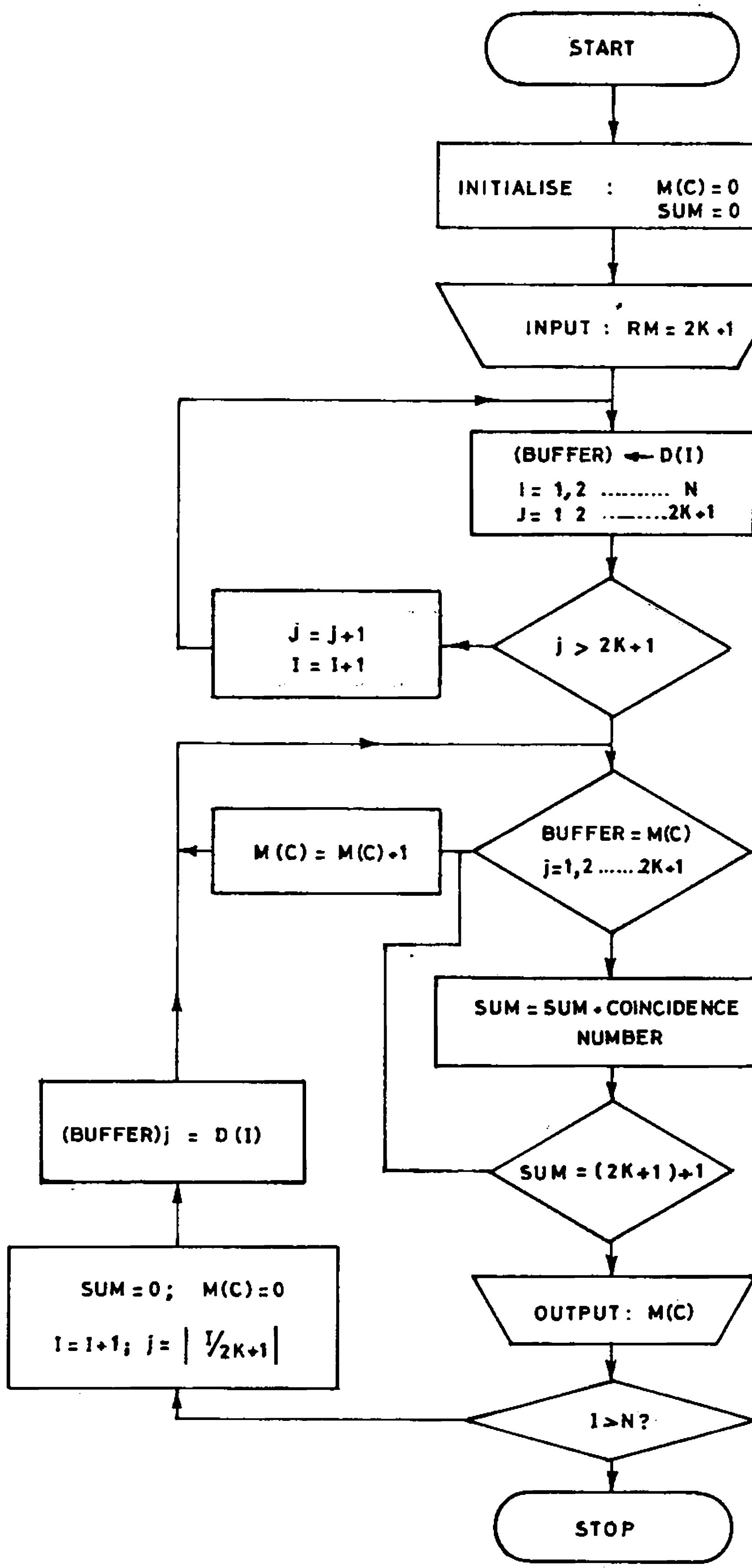

START
INITIALISE : M(C) = 0
SUM = 0
INPUT : RM = 2K +1
(BUFFER) ← D(I)
I = 1, 2 .... N
J = 1 2 ....2K+1
j > 2K+1
j = j+1
I = I+1
BUFFER = M(C)
j=1,2 ...... 2K+1
M (C) = M (C)+1
SUM = SUM + COINCIDENCE NUMBER
SUM = (2K+1)+1
OUTPUT : M(C)
I>N?
STOP
(BUFFER)j = D (I)
SUM = 0; M(C) = 0
I = I+1; j = | I/2K+1 |

Fig. 6·2

level of the input sequence and Tc is the median counter clock period. The MF flow chart is given in Fig.6.2.

The hardware for a 5 point median can be easily converted to that of a 3 point window. The necessary external changes are in the buffer to store the data in mod 3 form and Sum comparator value (K+1). Secondly for the n^{th} ranked non median operation the sum comparator value can be adjusted as per the requirement.

The MF hardware described shows flexibility in choosing the window size and the ranked operation. This flexibility is not available in any of the present hardware algorithms [10,11]. The selection network [10] fails to give a hardware structure for the window size beyond 7. Further, selection network is neither optimised to minimum hardware nor minimum delay. The method presented here has the minimum hardware. This method can outperform software median filter [14] when x_1 is represented in smaller number of bits. A comparison between the selection network and the present method is shown in Table VI.A.

Table VI.A

Window Size (2K+1)	Selection network		Present method	
	Comparators	Delays	Comparators	Delays
3	3	3	4	variable
5	7	5	6	"
7	11	9	8	"
:	:	:	:	:
(2K+1)	Not available	Not available	(2K+2)	variable

6.2 VLSI Implementation:

The comparator method described in Fig. 6.1 though reduces the hardware does not reduce the comparison time inspite of its flexibility in obtaining the running median for different windows. In practical on-line applications, the delay in sorting the sequence is much more important than the hardware complexity. In this section MF realisation with unit delay is described. Let the input sequence be quantised to b bits. It is necessary to compare $(2^b - 1)$ levels to obtain the median. A method is described here to realise the median filter.

The median is the mid value of the input samples when they are ranked. Let the input sample be represented in b bits and a moving window (2K+1) be selected. Initially (2K+1) samples are latched and fed to the input port A of the comparators. The data to be compared are fed to B port of the comparator. Since the input data are represented in b bits, it may take any of the 2^b possible levels. The B inputs of the comparators is also represented in b bits and their value thresholded to $(2^b - 1)$ distinctive levels. Let these be represented by T^i.

The output $y_{(m)}$ at position m of a MF with a window (2K+1) is

$$y_{(m)} = \sum_{i=1}^{q} T^i (m) = \Phi \left[x_{(m-k)} \quad \ldots \; x_{(m)} \quad \ldots \; x_{(m+k)} \right]$$

where $q = 2^b$. The input sequence is compared with all the thresholded values T^i. The comparator output $A \geq B$ is passed

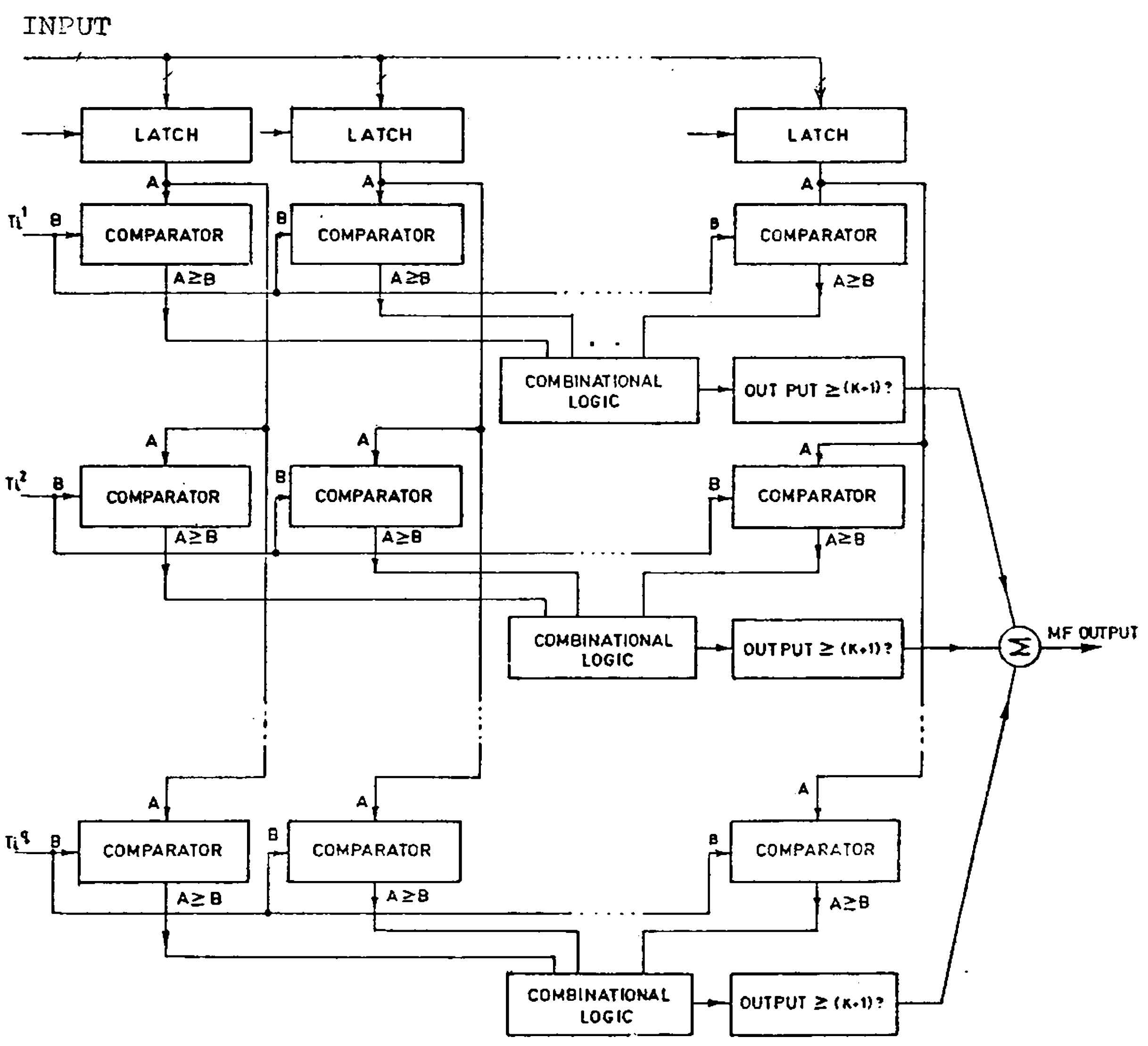

INPUT
LATCH
LATCH
LATCH
A
A
A
Ti1
B
COMPARATOR
B
COMPARATOR
B
COMPARATOR
A≥B
A≥B
A≥B
COMBINATIONAL LOGIC
OUT PUT ≥ (K+1)?
A
A
A
Ti2
B
COMPARATOR
B
COMPARATOR
B
COMPARATOR
A≥B
A≥B
A≥B
COMBINATIONAL LOGIC
OUTPUT ≥ (K+1)?
MF OUTPUT
A
A
A
Tiq
B
COMPARATOR
B
COMPARATOR
B
COMPARATOR
A≥B
A≥B
A≥B
COMBINATIONAL LOGIC
OUTPUT ≥ (K+1)?

Fig 6.3

through a combinational logic to check whether the largest (K+1) samples are equal to or greater than the threshold level. The summed output of the thresholded branches gives the true median value. A block diagram of this structure is shown in fig. 6.3.

The input samples are latched either by shift registers or latches. The input sample data are latched in modulo (2K+1) so that only the earliest data are replaced by the latest entry. These (2K+1) samples are compared with all possible q levels. A comparator has two input ports and an output port. The input sample is fed to the A input of comparator while the B input is the thresholded level to compare the input sequences. All the (2K+1) comparator outputs $A \geq B$ are passed through a combinational logic circuit to get the majority output (Ø or 1). The combinational logic circuit may be realised with standard logic gates or a ROM. The output of the combinational logic (1 bit) is summed using standard adders. The output gives the median of the input sequence at any position m. It may be noted that there are (q-1) identical blocks with the same A inputs. The B input is progressively increased from 1 to q. This method is suitable for VLSI implementation because all branches of q are identical. It is possible to modify the structure for n^{th} ranked operation. This will be useful for spatial normalisation like underwater target detection [35] where the operation needs higher degrees of freedom in choosing the normalisation algorithm to suit the environment. The n^{th} ranked operation can be obtained with slight change in the combinational logic or in the ROM table.

Major advantages of the threshold decomposition structure are: (1) Fastest on-line MF (2) Flexibility of realising n^{th} ranked operation (3) Filter designed for a given window can be easily modified for lower order window (4) structure has (q-1) identical blocks and easily implementable in VLSI.

B. Median Filtering for underwater target detection

Noise in the ocean is a superposition of anisotropic noise field due to rough surface and of an isotropic noise field in the absence of radiation from the surface. Detection process is complicated due to the details of the interference environment not being known. From the incoming signal a target "present or absent " decision is to be made by comparing each range cell voltage to a fixed threshold. This threshold value is a function of interference and receiver noise. The design of constant false alarm rate (CFAR) processor is achieved only if proper threshold value can be set for each range cell.

The detection process is simplified if the p.d.f's of signal and noise are known. In many cases we do not have complete apriori knowledge of these two. Hence detection of signal usually involves comparison of statistics based on the ratio of probability density function of signal plus noise and noise only conditions. Added to this underwater noise is not stationary. In order to overcome these, normalisation preceeds detection process. Once normalisation is carried out, the detection perofrmance should depend only on signal to noise ratio irrespective of the environmental changes and of signal or noise levels in the medium.

6.3 Ranked CFAR Processor:

Unknown level CFAR processing through cell averaging has been suggested by Weiss, Hansen, Trunk and Nitzberg [28-33]. Here it is necessary to estimate the possible interference noise power for each range cell.
This estimation r_K is achieved by considering K neighboring cells on either side (window size 2K+1). This estimate is

$$r_{ki} = \sum_{-K}^{K} r_i \quad i=1 \; .... \; L \qquad ..... \qquad (6.2)$$

Every time the window is moved by one cell, r_{ki} is reestimated.

For each position of i, the difference $(r_i - r_{ki})$ is computed.

This is the required normalised input to set detection threshold. The normalised output variance is always lower than the input signal variance and tends to match the actual signal variance as the window size increases. As K increases the r_{ki} tends asymptotically to an unbiased estimate.

A running median CFAR processor is shown in fig.6.4. The input to the delay line is in natural order. The median scanner quickly finds the median for a given window size. This can be done either in hardware [10,34] or in software [14]. Median output is now taken as an estimate of noise power. Variance with different window sizes are computed and shown in fig.6.5. It is observed that the median output variance σ_M^2 is

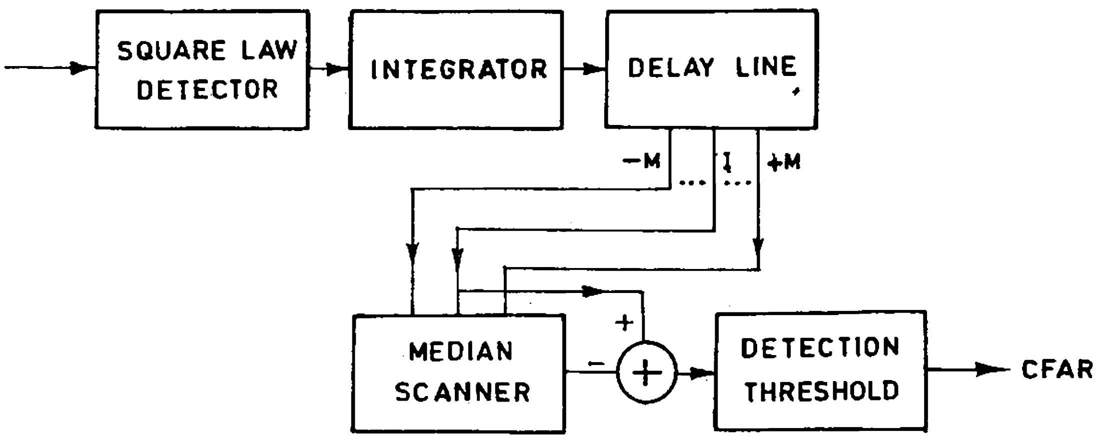


Fig 6.4 CFAR PROCESSOR

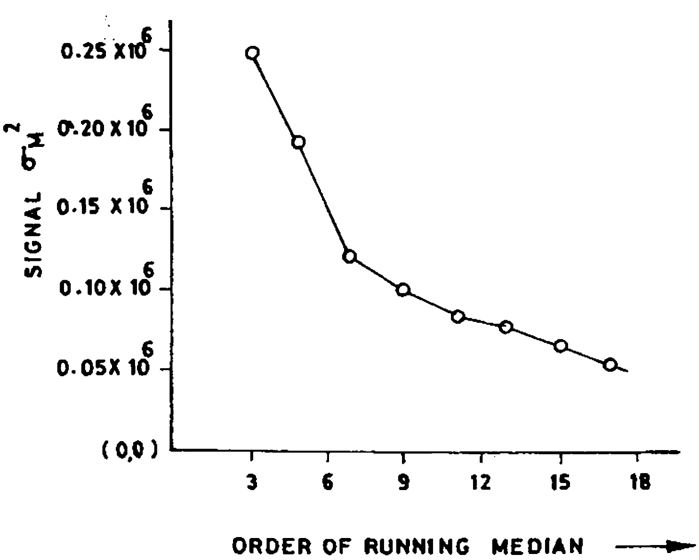


Fig 6.5 RUNNING MEDIAN Vs SIGNAL VARIANCE

always larger than the variance of the sample mean for gaussian noise. That is

$$\sigma_M^2 = C_K^2 \frac{\sigma^2}{(2K+1)} \qquad \ldots\ldots \quad (6.3)$$

where $C_K \geq 1$. Cadwell [26] proposed an approximation for the value C_K.

$$C_K = \sqrt{\frac{\pi}{2}} \left[1 - \frac{(4-\pi)}{2(2K+1)} \right] + O\left[\frac{1}{(2K+1)^2} \right] \qquad \ldots\ldots \quad (6.4)$$

It is observed that σ_M^2 reduces at a faster rate upto the window size of 7 and then gradually decreases thereafter.

Performance curve:

The normalised variance power versus window size is shown in fig.6.6 for running median and moving average. Simulation was carried out for radiated noise of the target with interference noise having normal distribution. It was observed that the running median and the moving cell average methods behave in a similar fashion as the window size increases. The normalising voltage is an estimate of noise variance and the error of this estimator decreases as the samples in the estimator increases.

The running median is slightly inferior to the moving average (MA) for a given window size. The variance is minimum only when it is computed with respect to its mean. The variance for any other value will be always higher. The performance of the running median can be improved by taking m^{th}

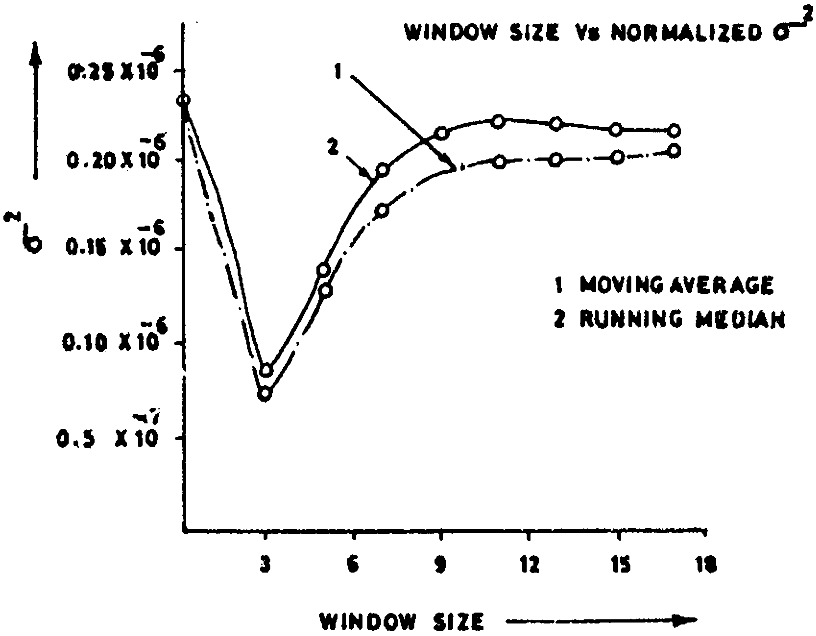

WINDOW SIZE Vs NORMALIZED σ²
0.25 X 10⁻⁶
0.20 X 10⁻⁶
0.15 X 10⁻⁶
0.10 X 10⁻⁶
0.5 X 10⁻⁷
σ²
1
2
1 MOVING AVERAGE
2 RUNNING MEDIAN
3
6
9
12
15
18
WINDOW SIZE

Fig 6.6

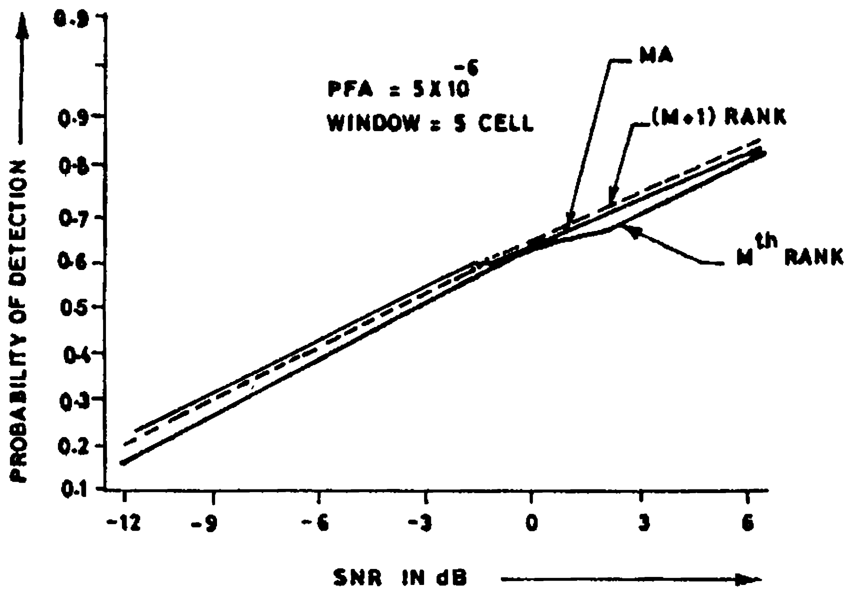

PFA = 5 X 10^-6
WINDOW = 5 CELL
MA
(M+1) RANK
M^th RANK
PROBABILITY OF DETECTION
SNR IN dB
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
-12
-9
-6
-3
0
3
6

Fig 6.7

ranked window sample instead of median sample. The ocean noise being an increasing function of sea-state. The noise estimate can be adapted for median or n^{th} ranked (say K or $K \pm 1$ ranked sample) operation when the medium is rough or SNR is low. Curves that give the probability of detection versus signal to noise ratio for an unknown level CFAR processor implemented by cell averaging and median filtering are shown in fig.6.7. These detection curves are for a false alarm probability 5×10^{-6} and window size five.

The running median normalisation provides a higher degree of freedom to choose the spatial normalistion from time to time. The improvement in performance is appreciable under weak SNR. Furtner this algorithm suits on-line implementation since it does not involve any complexity either in software or hardware.

C. **Picture Processing.**

Median filters have been used in picture processing mainly for impulse noise removal [3] and to some extent for the removal of salt and pepper noise [7]. Two dimensional median filters have been used for picture processing [11] and seperable MFs have been shown to have some advantages [16] for such cases. Tyan and Justusson [18] have proposed various 2D median windows for picture processing. These windows are basically symmetric around some prescribed axes and have been proved to reduce the image variance [11]. Narendra has established that the performance of a median filter is better than that of a linear smoother.

Applications other than smoothing are possible. If we consider the basic structure of a median filter, the output samples are a subset of the input samples with trend changeover (maxima/minima) samples being eliminated. In the place of such samples, the median filter substitutes one of the samples within a window. Thus seperable median filters perform identical to other types of 2D filters [7,11]. In general the output sequence has as many samples at the output that are correlated as those that are replaced, i.e. the output sequence retains the correlation continuity. This can be usefully exploited in picture processing.

If we consider a picture whose pixel values are available as a sequence of samples any hidden contours can possibly be brought out by obtaining the correlation function of the output sequence of a median filter. Further, the change in trend of correlation function yields information regarding the trend of the signal contours. An example of a picture 'Girl with hat on' is taken for this study. The picture is available in digitised form.

6.4 Feature extraction

If we consider a one dimensional running median output for each line of the picture, it can be seen that the line corresponding to equal luminance show uniform values of correlation. However, for a line encompassing the boundaries of objects with different luminance, the correlation varies (fig. 6.8). For unfiltered picture there may be fluctuations in the correlation. Since median filtering

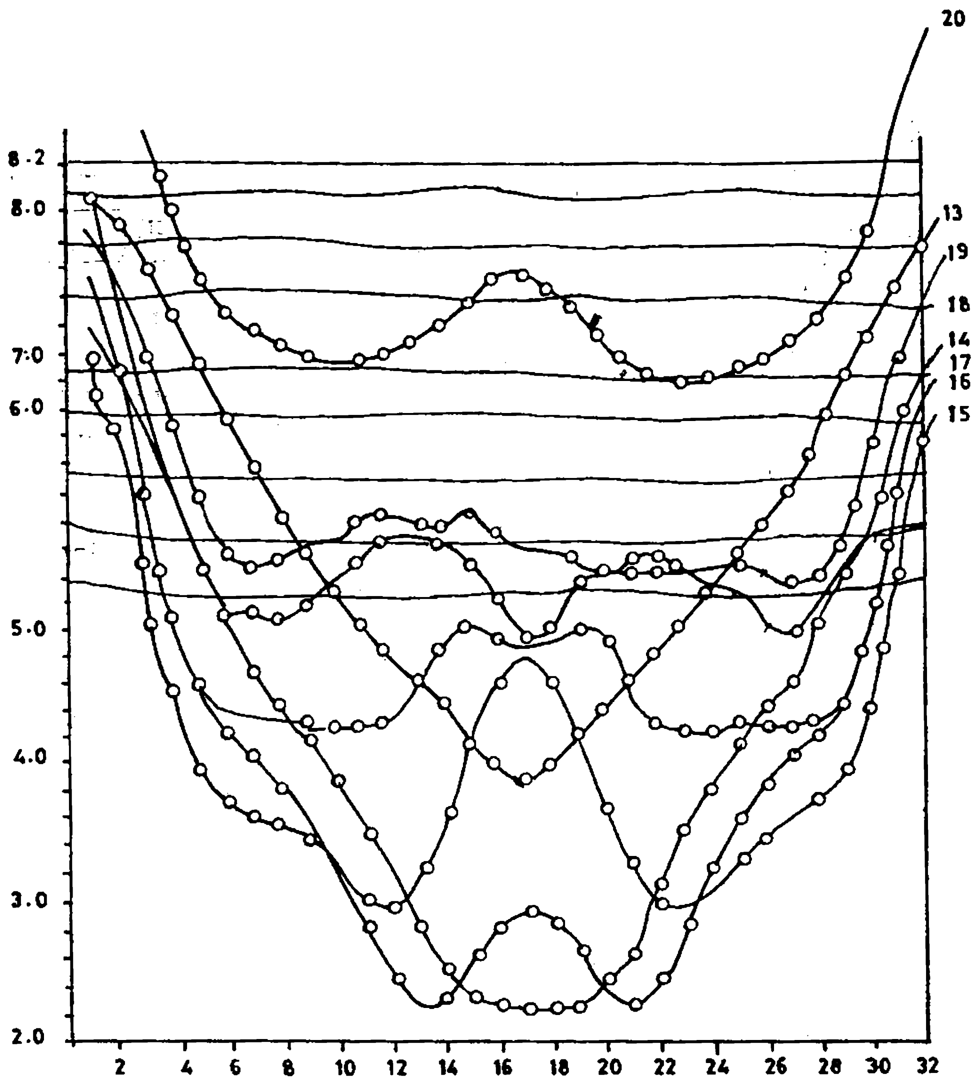


Fig 6·8 CORRELATION PLOTTED LINE BY LINE

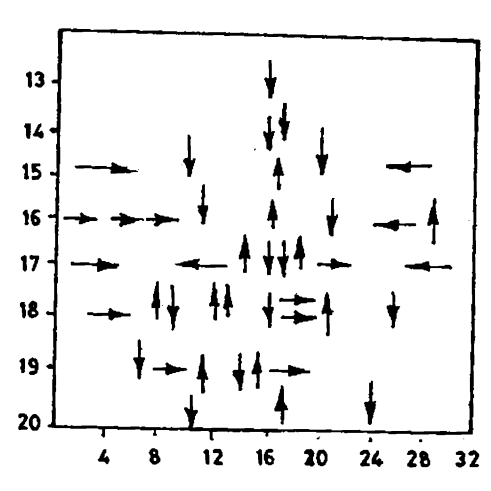


Fig 6·9

preserves edges and removes spikes, the correlation shows distinct regions of monotonic rise or fall.

A 3 point running median output is obtained for each line in this analysis.

$$Y_1(n) = \phi^1 \left\{ x_{(1,n-1)}, x_{(1,n)}, x_{(1,n+1)} \right\}$$

$$\vdots \qquad \vdots \qquad \vdots$$

$$Y_{32}(n) = \phi^{32} \left\{ x_{(32,n-1)}, x_{(32,n)}, x_{(32,n+1)} \right\} \qquad \ldots (6.5)$$

To obtain the correlation line by line, spectrum of each line is obtained for both MF output and input. The correlation in frequency domain is given by

$$Z^i(f) = X^{i*}(f) \cdot Y^i(f) \qquad \ldots.. \quad (6.6)$$

where $X^i(f)$ is the spectrum of the i^{th} input line and $Y^i(f)$ that of the i^{th} output line (* indicates complex conjugate).The pattern of the boundaries of different objects show different slopes in the correlation curves. The results of the correlation function for different lines of the picture is shown in a series of curves in fig.6.8. The picture lines 1 to 11 are of uniform luminance representing only a background. Similarly lines 20 onwards also show only a uniform background being the image of a single object. The intervening lines carry details of face, eye, and the side obscured by the edge of the hat.

The interesting result is the manner in which the correlation curve varies from line 13 to line 20. Line 13 is a 'V' shaped correlation curve. The dip occurs at a point approximately in the middle. The dip flattens out in line 14 indicating the onset of a different and a new object in this region. Line 15 shows the formation of an upward cusp in this dip. This decorrelation clearly indicates that an obscuring object is forming. Line 16 shows further strengthening of this cusp. This is followed by line 17 showing a symetrical flatening of both the top of the cusp and the peak correlation of the adjoining regions. Line 18 completes the definition of this second object by starting to show a decrease in the value of the correlation to be followed by a complete flattening of the curve to line 19. That is, the presence of a different object between vertical position 14 to 18 and horizontal 13 to 22 is highlighted by the correlation of the median filtered output.

If the changes in the slopes of the correlation are plotted for these lines as in fig.6.9 with arrows indicating the trend instead of the actual values, the existence of the obscuring object as well the hidden object contour can be very clearly interpreted.

It is possible to define and extract features from pictures by studying the correlation functions of the median filtered versions. It is easy to see that any change in the trend of the correlation indicates the existence of a unique feature. However this can be concluded only after inspecting the one dimensional median filtered output line by line. When 2D filter is employed the uniqueness of the feature may be lost if the window is not small enough. The conclusions drawn from one dimensional output are applicable to 2D images also since MF's are seperable. Thus

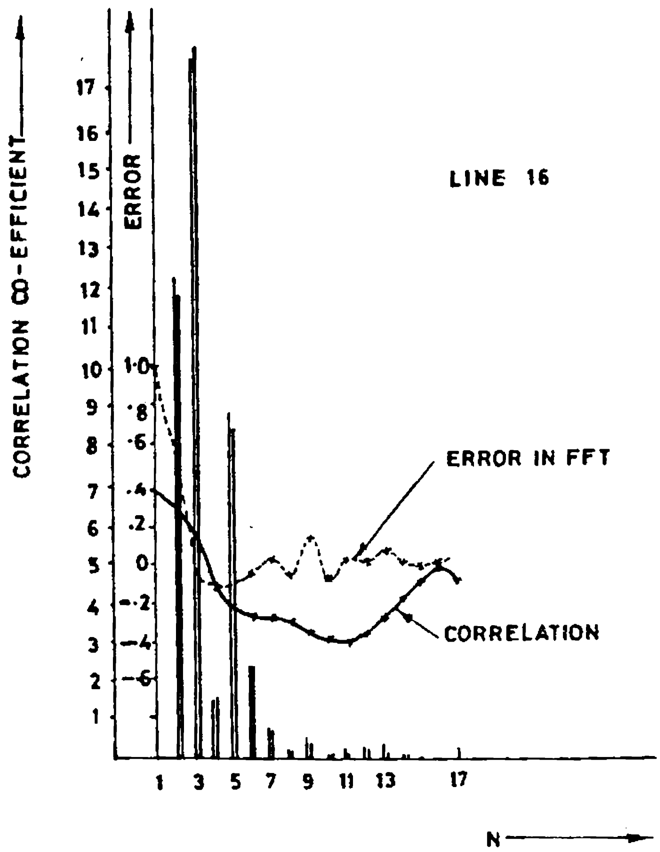


Fig 6.10 (a)

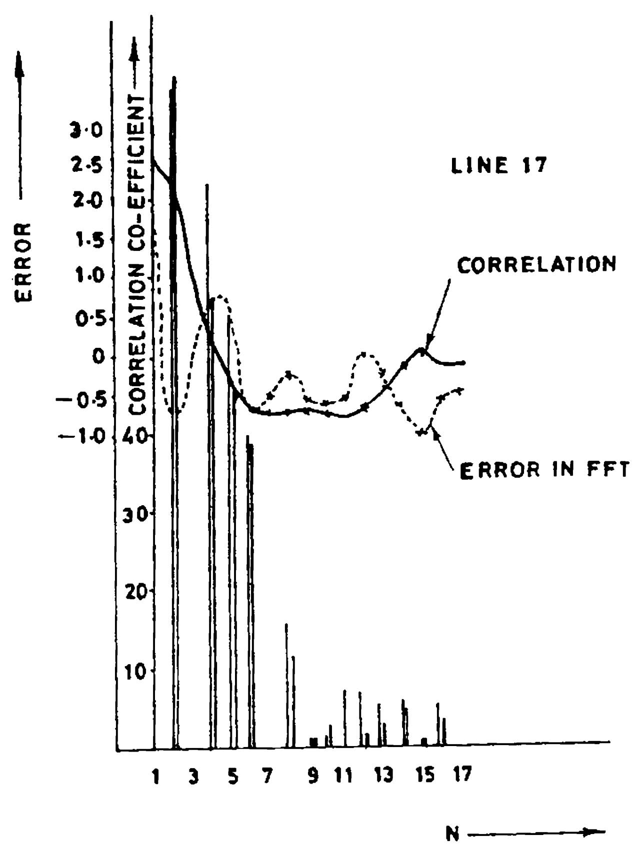
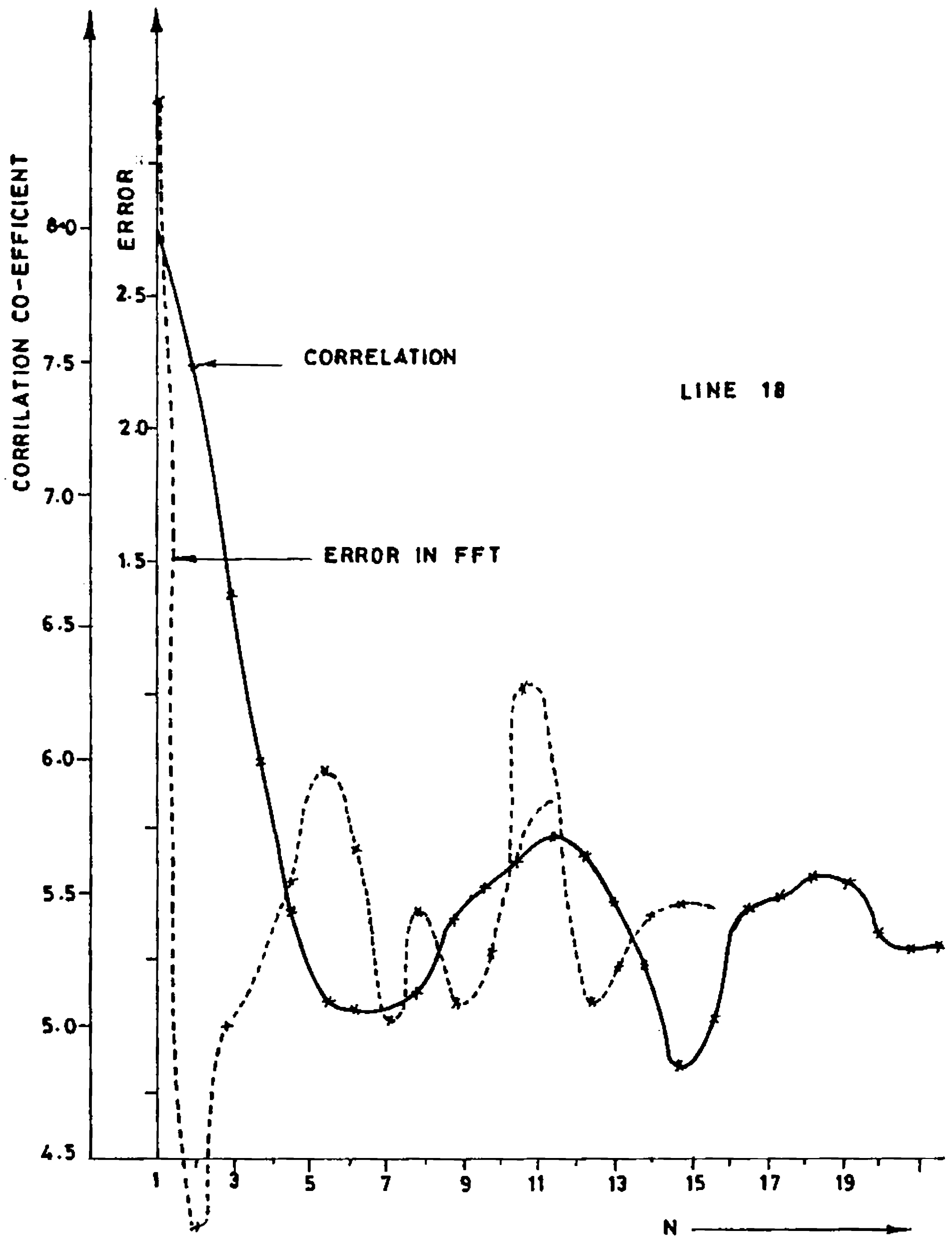


Fig 6.10 (b)



CORRILATION CO-EFFICIENT
ERROR
8·0
7.5
7.0
6.5
6.0
5.5
5.0
4.5
2.5
2.0
1.5
CORRELATION
LINE 18
ERROR IN FFT
1
3
5
7
9
11
13
15
17
19
N

Fig 6·10(c)

he number of features that are present in a picture can be found y counting the number of trend changes in the median filtered ine by line correlation function. Extraction of the actual eatures is slightly more complicated. For this, it is necessary to go back to part of the original image which is lemarcated by regions of change in correlation and reproduce those portions as unique features. However, if these features are repeated elsewhere in the same picture with a different value of correlation, then this method does not indicate the sameness and would classify them as a different features.

Now that it is well known that the Fourier spectrum is a good measure of features, the effect of median filtering on the FFT of the picture was studied. Fig.6.10 shows the plot of error in FFT and correlation for lines 16, 17 and 18 of the picture. The monotonic regions in correlation correspond to large error in the FFT. This error resembles a damped sine wave, whereas the region potentially capable of representing features (random fluctuations in the FFT) should show negligible error. This is clearly indicated in fig. 6.10(a), (b) and (c). Thus the correlation function of the MF output serves as an indication to the existence of hidden contours. It is possible to use correlation technique to identify the existence of features in an image.

D. Speech Processing:

It is well known that the pitch period of speech can be estimated by first center clipping the samples and then examining the auto correlation function. When speech is passed

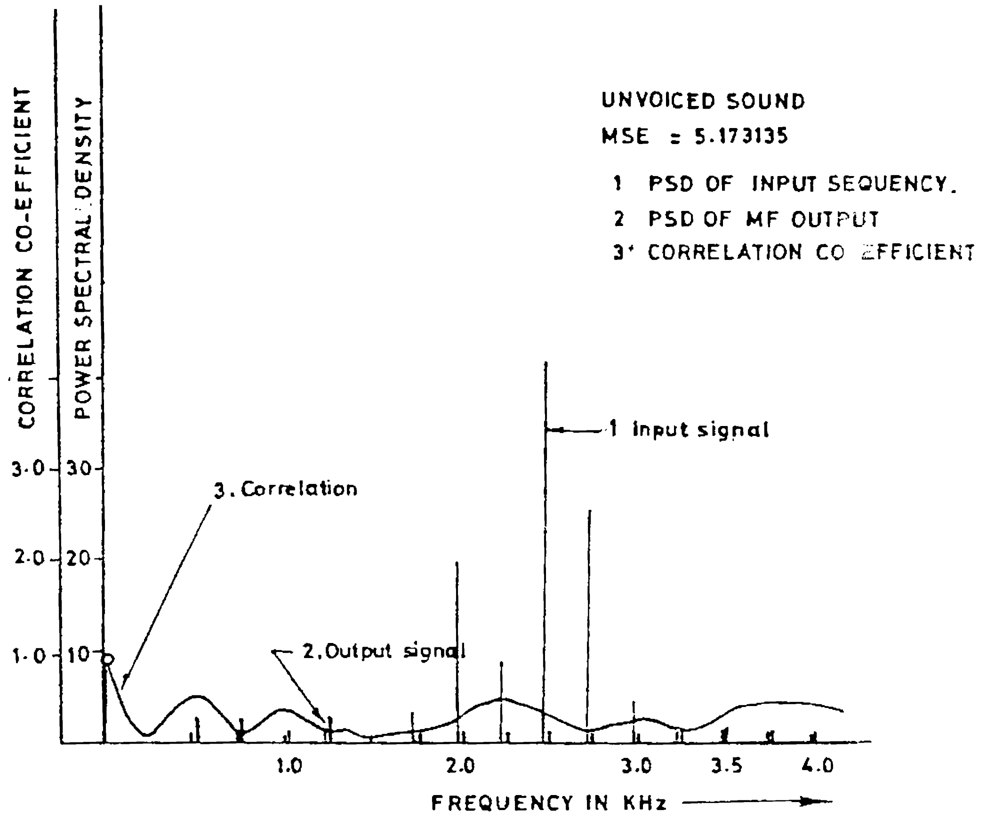


Fig 6.11(a)

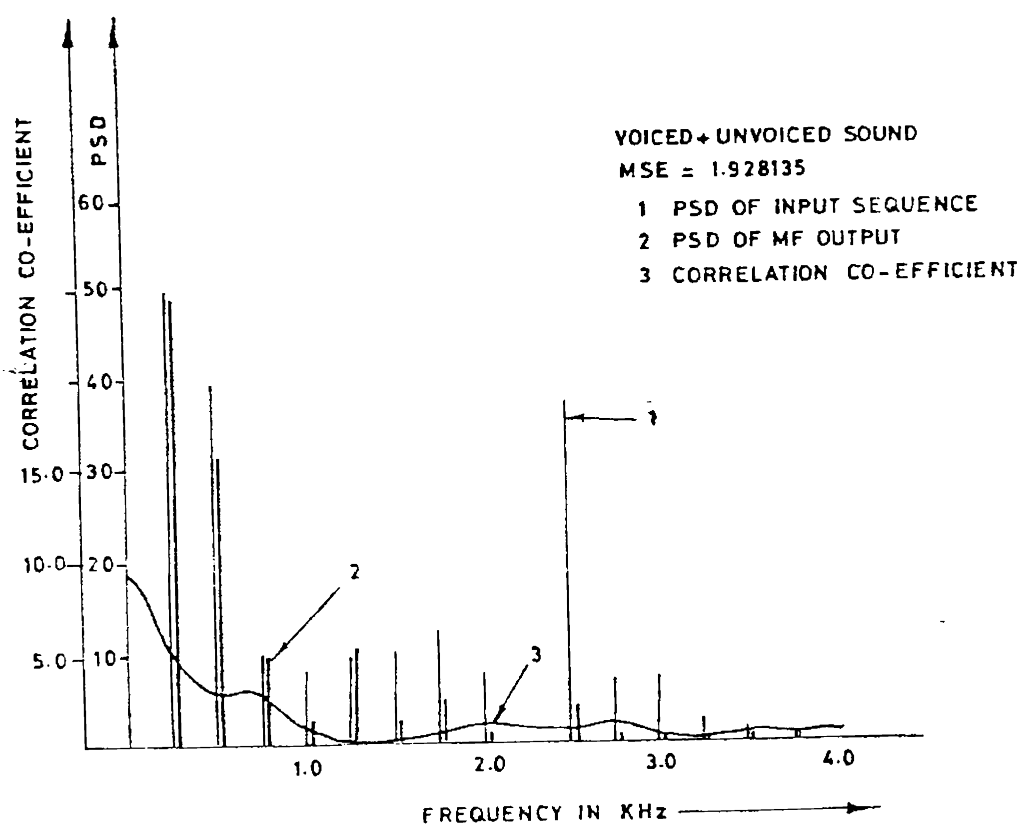


Fig 6.11(b)

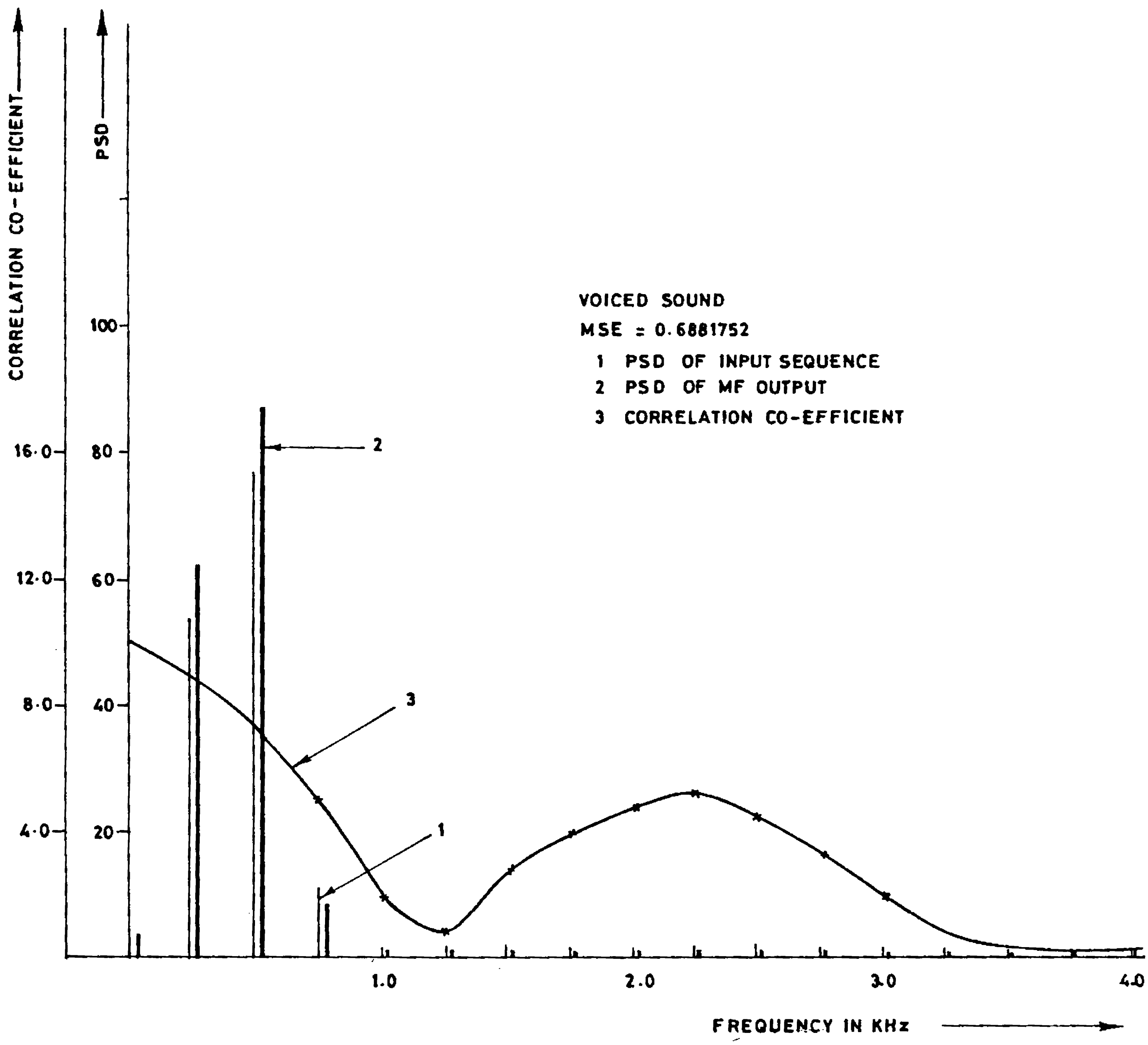

CORRELATION CO-EFFICIENT
PSD
VOICED SOUND
MSE = 0.6881752
1 PSD OF INPUT SEQUENCE
2 PSD OF MF OUTPUT
3 CORRELATION CO-EFFICIENT
100
16.0
80
12.0
60
8.0
40
4.0
20
1
2
3
1.0
2.0
3.0
4.0
FREQUENCY IN KHz

Fig 6.11(c)

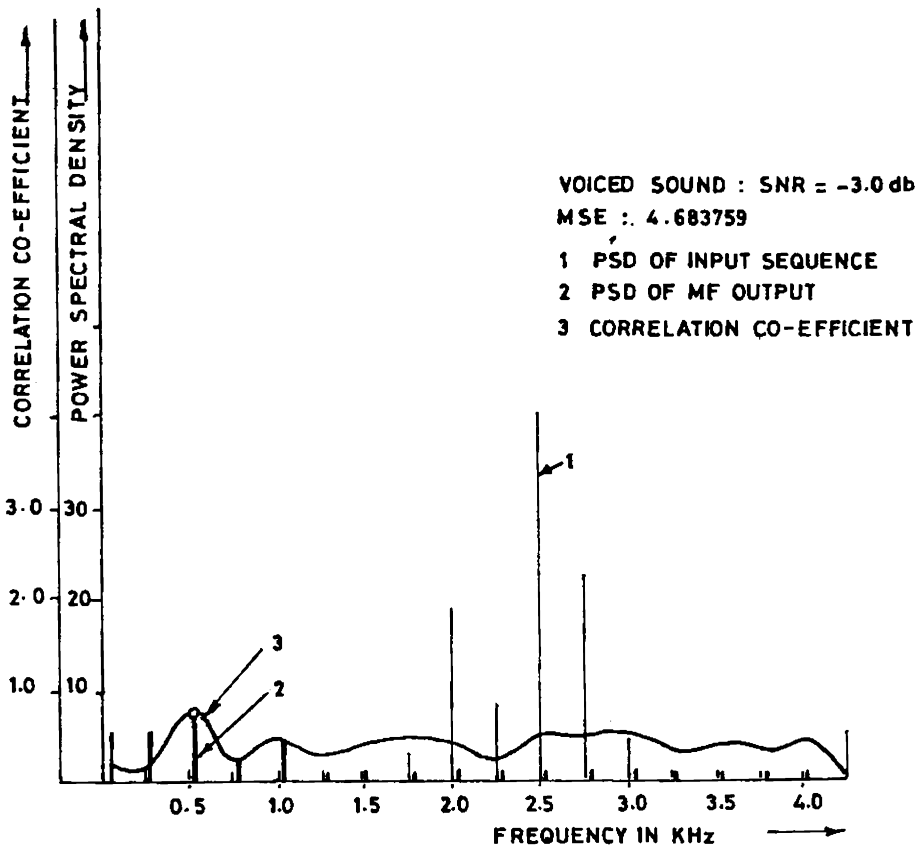


Fig 6.12 (a)

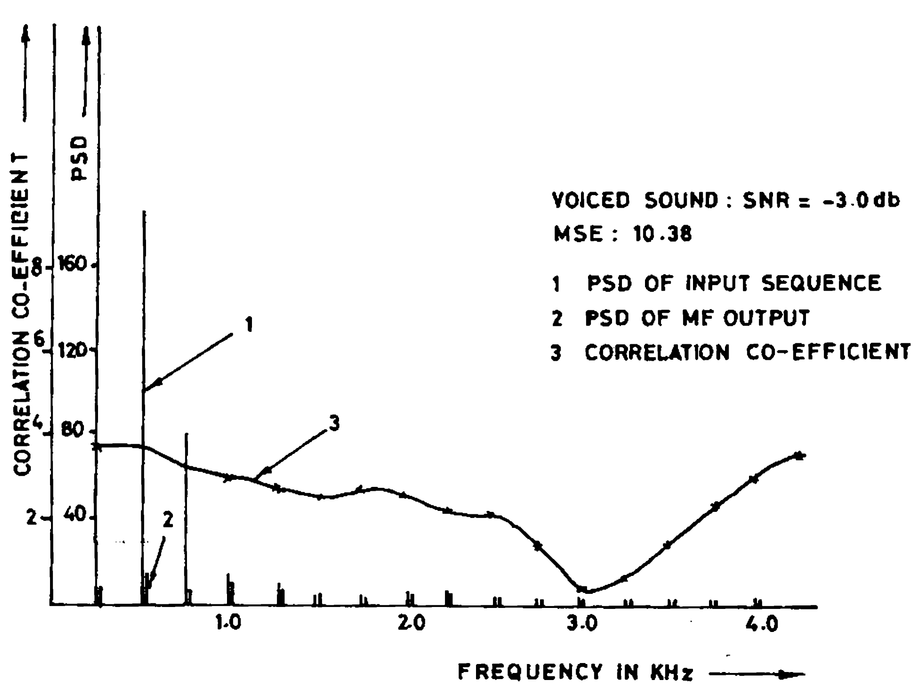


Fig 6.12 (b)

through a MF, a good number of the samples remain unchanged. Thus most of the characteristics of the auto correlation function are preserved. Using this fact the autocorrelation function of median filtered speech samples were examined.

The speech data is made up of segments consisting voiced, unvoiced and boundary between unvoiced and voiced sounds. The Fourier transform of the MF output when compared with the input signal presents some interesting results. The MF effect on unvoiced signal is studied first. Fig.6.11(a) shows a typical unvoiced portion in frequency domain. The input signal shows strong line component at 2KHz, 2.25KHz, 2.5KHz and 2.75KHz. The MF output does not show these components at all. That is the MF output destroys all dominant frequency components.

The boundary portion containing voiced and unvoiced portion of the signal is passed through the MF. The MF output spectrum removes the unvoiced component while retaining the 250Hz component belonging to the voiced sound. This is evident from the fig.6.11(b). The MF retains all the voiced frequency components as shown in fig. 6.11(c) for voiced segment.

When the speech signal is corrupted by Gaussian noise, the frequency domain plot is given in fig.6.12. The MF output still retains all the dominant line components of voiced sound. This indicates the retention of certain characteristic frequencies which may be formant frequencies. The MSE between the input and output spectrum shows a related behaviour. The MSE is very small for voiced portions while it is high for unvoiced portions. Since there is a marked difference between the MSE's for voiced and unvoiced portions, this can be used as a measure of determining the voiced/unvoiced boundaries and regions. Correlation function plots for these regions show that the voiced signal is strongly correlated whereas it is not so for unvoiced signal.

6.5 Conclusion and discussion

The hardware realisation of Mf with minimum hardware has a good potential in constant false alarm rate receiver. This is mainly due to two reasons viz. (1) flexibility in changing the window size (2) ranked order filter. The VLSI implementations have additional applications in speech and Image processing. The unit delay time in selection and sorting of samples is also attractive for image processing applications.

The running median normalisation for an unknown level CFAR processor is found useful for weak signal detection. The weak signal detection capability is enhanced by choosing n^{th} ranked output for spatial normalisation. The implementation does not involve any complexity either in software or hardware. The correlation method of MF for feature extraction in image processing, identification of voiced, unvoiced sound and extraction of formant number is encouraging.

Chapter VII

CONCLUSIONS

Detection of edges in images, filtering techniques for preserving sharp discontinuities in signals and reducing computation are some of the attributes one looks for in techniques of Image and Signal processing. As discussed in Chapter II, MF seems to provide possible answers for these requirements. However due to their inherent non-linear nature, straight forward analysis of median filters like linear filters have not been possible. This thesis has attempted such an analysis from several angles.

To provide a frame-work for analysing MF the chapters III and V introduced certain new concepts in MF viz. transformation matrix, Median Matrix and Column sum. The approach using the weighting matrix, modified weighting matrix and ultimately the median matrix is entirely new and is not based on any work previously reported in MFs. It has been established that a median matrix operation can be defined for MF and that for a class of signals (monotonic, periodic etc.) the median matrix can be written by inspection. The MF output for these class of signals can also be written down by inspection, which inturn means that this algorithm can give the MF output without actually performing any comparison. Attempt to extend this to general class of signals has not succeeded. One reason is the difficulty of codifying statistical properties of the input into coefficients of the weighting matrix. This area is open for further work and quite possibly, approximations to the exact median matrix can be derived by relating the position and distribution of 1's in the [M] to the probability distribution function of the signal.

An entirely new concept - namely the column sum of [M] has been defined While indicating the number of times a particular sample maps onto the output, this parameter proves to be of interest for the study of signal properties. Further indications as to the number of roots and root paths possible are also indicated by the pattern of column sums. The results presented represent the analytical elegance of this new concept. It has not been possible to relate this to the design of MFs or, in general, to performance specifications of MFs. It should be noted at this stage that others[20,25] also have evaluated the number of roots for a MF They have neither established any relationship between roots and the design of MF nor have specified the use of roots in the analysis of MF. The results presented in this thesis differ from those of the rest both in the approach and values. In this approach as explained in Chapter III, each root is described in terms of the sample that are being repeatedly mapped onto the output and not the sample values themselves. The determination of the number of roots itself has proved to be a good exercise in combinatorial arithmetic and provided the surprising result that the number of invalid paths at the n^{th} column is simply (2K+1) times the number of valid paths at the $(n-3)^{rd}$ column.

Based on the analysis provided for column sum and the tree diagram for the signal state description has been attempted for the column sum patterns. The results are interesting to the extent that a behaviour similar to that of binary signal states is exhibited by the column sums It has not been possible to see why sample values as well as their appearance numbers at the MF output should behave in similar fashion. It is felt that there is some scope for further analytical work in this area

While analysing the column sum patterns, it has been observed that the state description and the table for evaluation of valid paths and roots bear a strong resemblence to the methods used in logic circuit design. Though the results presented in this work are complete as far as the objectives are concerned, it would be very useful to extend the method of prime implicants to this problem, particularly when cases of very long column lengths and large window widths are concerned.

Two new methods namely Fast Convergence Median Filter and Interpolated Median filters are defined. The results show that these two filters are very powerful for on-line image/picture processing. It has been shown in Chapter IV that there is considerable savings in computation due to their simplicity while processing the images. Further the performance is on par with seperable median filter and better than averaging filter for both white and Gaussian noises

Very few authors have tried to analyse the MF in frequency domain for the reason that MF is non-linear. The work in Chapter V is an attempt to characterise certain class of MFS in frequency domain. The approach used here is unique in that the difference between the input-output sample contribution to DFT and the effect of periodicity on the MF output have been studied. The results are not complete in several aspects and it has not been possible to generalise the input-output relationship in the frequency domain. On the contrary it is felt that an approach through sample number, similar to wave number, may lead to analysis methods more useful for the design of MFs.

On the practical side, improved realisation of MF has been presented in this work. It has also been shown that VLSI implementation of the MF will lead to more efficient systems. While discussing n^{th} ranked order realistion, which will prove very useful for sonar applications, it is felt that FCMF and IMF may prove useful under certain conditions. The results are presented in Chapter VI.

As mentioned elsewhere in the thesis, the output of the MF contains only samples of the input and no new samples are generated. This essentially means that the output and input are correlated to the extent determined by the behaviour of trend changes in the signal. This has led the author to investigate the relationship between the correlation function and the MF filtering. This approach is a useful application in Image Processing and possibly in Pattern Recognition. Similarly its usefulness in Speech Processing is also demonstrated.

Concluding the thesis, MF input-output characterisation has been defined and modified to get median matrix which is useful to extract several properties of a input sequence. Characterisation of MF in the frequency domain for deterministic signals has been attempted. Concepts of FCMF and IMF have been introduced and their application to image processing has been discussed. A simple and flexible MF filter hardware has been developed and applied to underwater target detection.

REFERENCES

1 J.W. Tukey, "Non linear (non superposable) methods for smoothing data", Int Conf. Rec., 1974, EASCON P.673.

2 L R. Rabiner, M R. Sambur and C.E. Schmitt "Applications of a nonlinear smoothing algorithm to speech processing", IEEE trans. Acoust., Speech. Signal Processing, Vol. ASSP-23, pp 552-557 Dec. 1975.

3 B R. Frieden, "A new restoring algorithm for preferential enhancement of edge gradient", Jour. opt Soc. America, vol.66 pp 280-283, 1976.

4. N.J Jayant "Average and median based smoothing techniques for improving digital speech quality in the presence of transmission errors. IEEE trans. Commn. vol. COM-24, pp 1043-1045 Sep. 1976.

5. R. Steele and D.J. Goodman "Detection and selective smoothing of transmission errors in linear PCM, "Bell system Tech. Journal, Vol.56, pp 300-409, Mar. 1977.

6. J.W. Tukey "The ninther a technique for low effect robust (resistance) location in large samples", in David H.A., ed. contributions to "survey sampling and applied statistics" Academic Press, pp 251-257, 1978.

7. W.K. Pratt, Digital Image Processing, Wiley interscience, New York, NY 1978.

8. S.G. Tyan, "Fixed points of running medians", Dept. of Elect Engg & Electro Physics, Polytechnic Inst. of New York, Brooklyn, N.Y. 1977.

9. B. Justusson "Noise reduction by median filtering", Dept.

of Maths Royal Inst of Tech , Report No.TRITA-May 1978-7 Stockholm, Sweden, 1978

10. M.I. Shamos,"Robust picture processing operators and their implementation as circuits, Proc. of Image understanding Workshop, Pittsburg, Penn., pp 127-129, Nov. 1978.

11. T.S. Huang, G.T. Yang and G.Y. Tang, "A fast two dimensional median filtering algorithm", IEEE trans. Acoust, Speech, Signal processing, Vol.ASSP-27, pp 13-18, Feb. 1979.

12. V.K. Aatre, E. Ataman and K.M. Wong, "Median filtering", in Proc. 17th Ann. Allerton Conf. Commn, Contr. Comput., Oct. 1979, pp.886-895.

13. E. Ataman,V.K. Aatre and K.M. Wong, "Some statistical properties of median filters", Dept. of Elect. Engg., Nova Scotia Tech. College, Halifax, Nova Scotia, Dec. 1979.

14 E Ataman V.K. Aatre and K.M. Wong "A fast method for real-time median filtering", IEEE trans. Acoust., Speech, Signal processing, vol.ASSP-28, pp 415-420, Aug. 1980.

15. Vellman A.F., "Definition and comparison of robust nonlinear data smoothing algorithms", Jour. Amer. Statist. Ass., pp 609, Sept 1980.

16. P.M Narendra "A seperable median filter for smoothing", IEEE trans. pattern Analysis and machine intelligence, Vol PAM 1-3, pp 20-29, Jan 1981.

17. N.C. Gallagher and G.L.Wise, "A theoretical analysis of the

filters", Signal processing, June 1984, pp 67-76, North Holland.

25. J. Patrick Fitch, E.J. Coyle and N.C. Gallagher, "Root properties and convergence rates of median filters", IEEE trans. Acoust., speech and Signal processing, Vol.ASSP-33, pp 230-239, Feb. 1985.

26. J.H. Cadwell, "The distribution of quantities of small samples", Biometrika, vol.39, pp 207-211, 1952.

27. H.A. David, "Order Statistics", John Wiley & Sons, 1970.

28. Roman Nitzberg, "Constant false alarm rate signal processors for several types of interference", IEEE trans., Aerospace and Electronics Systems, Vol.AES-8, pp 29-34, Jan. 72.

29. H.G. Hansen and H.R. Ward, "Detection performance of the Cell averaging Log/CFAR receiver", IEEE trans. Aerospace and Electronics System, vol.8, pp 648-652, Sep. 1972.

30. H.G. Hansen, "Constant false alarm rate processing in search radars", in Proc. IEE, int. Radar Conf., London, pp 324-332, Oct. 1973.

31. K. Kendall and A. Stuart, "The advanced theory of statistics", vol.I, Macmillan, New York, N.Y. 1977.

32. G.V. Trunk, "Range resolution of targets using automatic detectors", IEEE trans. Aerospace and Electronics system, vol.AES-14, pp 750-755, Sept. 1978.

33. M. Weiss, "Analysis of some modified call averaging CFAR processors in multiple target situation", IEEE trans.

Aerospace and Electronics Systems, vol.AES-18, pp 102-114, Jan 1982.

Publication from the thesis

34. G. Devarajan V.K. Aatre and C.S. Sridhar "Median Filtering: Certain properties and real time implementation" IEEE Int Conf. Computers, Systems & Signal Processing, Bangalore, India, 1984.

35. G. Devarajan V K. Aatre and C.S. Sridhar "Application of Median filtering for underwater target detection" IEEE Int Conf Computers, Systems and Signal Processing Bangalore India, 1984.

36. G. Devarajan, V.K. Aatre and C.S. Sridhar, "Feature extraction using median filtering techniques", SPIE Int. Symposium on Pattern Recognition and Acoustical Imaging, California Feb 1987.

37. G. Devarajan V.K. Aatre and C.S. Sridhar, "Evaluation of number of roots in Median filters" (to be communicated).

38. G. Devarajan, V.K. Aatre and C.S. Sridhar, "Analysis of median filters", communicated to IEEE trans. Acoust., Speech Signal Processing.