# COMPUTATION OF A CLASS OF DETAIL PRESERVING FILTERS IN PIPELINE ARCHITECTURES

Amelia Fong

Department of Computing and Information Science
University of Guelph
Guelph, Ontario
Canada N1G 2W1

**Abstract**

Recently a new class of non-linear hybrid filters which combine FIR substructures with median operations were proposed and analyzed. They are shown to have excellent detail preserving properties and are computationally more efficient than conventional median and K-nearest neighbour averaging filters. In this paper, we propose a class of parallel algorithms for their computation using pipeline architectures. We described the algorithms for different hardware configuration, i.e. for the case where one comparator is available and for machines with two comparators. Analysis of the algorithms for both cases are also given. It is shown that using a commercial pipeline processor which can perform each processing pass at video refresh rate (30 frames per sec), except for the three-level bidirectional hybrid filter, all the filters can be computed using the proposed algorithms in less than one second. The minimum number of refresh memories needed without having to perform image transfer to and from disk are also derived and its implications for the choice of algorithms are discussed.

**Keywords:** Image enhancement, non-linear digital filters, parallel computation, pipeline architectures.

## I. Introduction

A fundamental problem in image restoration is to remove the additive noise produced by the imaging system without blurring the fine details of the image. This problem arises in e.g. machine vision, computer tomography and in other X-ray imaging system [RNR, 1984]. One major class of filters for image enhancement is based on ranked order statistics [BHM, 1983]. The best known of these is the median filter, whose statistical properties are well understood [AAW, 1981], [AG, 1982], [GW, 1981], [NG, 1984]. Two dimensional median filters can smooth noisy images while retaining the edge structures almost intact [AC, 1984], [SF, 1985], [NG, 1983]. However, median filters have poor abilities to retain fine details like lines.

Recently, a new class of nonlinear filters called FIR-Median Hybrid filters (FMH filters) was introduced in [NHN, 1987]. Also, a multilevel operation that makes it possible to build multilevel FMH filters that retain details of the image irrespective of their orientation was also introduced. It was shown also that the FMH filters preserves more subtle details than the median and K-nearest neighbor averaging (KAVE) filters [DR, 1978]. The FMH filters also preserve edges in noisy images better than the median and the KAVE filters, see [NHN, 1987].

The main problem with non-linear filters involving ordering is that they are very computationally intensive especially when higher order filters or large windows are used. In many real time applications, such as automated visual inspection where speed requirement is important, techniques for fast computation of image processing filters are of great interest. In these type of applications often parallel architectures are used. Recently there has been many new algorithms developed for pipeline architectures. [S, 1988], [SD, 1987], [SDP, 1988], [DF, 1988]. Often, we must rethink traditional approaches used for sequential computers to take advantage of the parallelism. In this paper we propose a technique for computing the class of FMH filters in a pipeline image processing system equipped with hardware comparators. Section II describes the class of algorithms. These include algorithms for the class of unidirectional, bidirectional and the multilevel FMH filters. Section III gives the algorithms for the above three classes of FMH filters if multiple processing units are available.

Section IV contains analysis of the algorithms and its implication on the choice of algorithms based on the resource available. Section V concludes the paper.

## II. The Algorithms

The architecture we are assuming is a pipeline architecture such as in [SD, 1987], [DF, 1988]. A schematic diagram is shown in Fig. 1. There are a number of refresh memories or image buffers. We assume that that the output from the image buffers can be shifted horizontally and/or vertically. We further assume that one of the pipeline processing stage is a hardware comparator. Most of the commercially available general purpose pipeline architectures are equipped with such comparator. For example, the DeAnza IP8500 has two such comparators which can be set independently to output either the maximum or the minimum of the two input pixel values.

In standard median filters, the median is taken over all samples inside the window. However when the number of samples is large, the ordering procedure is computationally intensive. In [NHN, 1987] a class of new filters called FMH filters are defined, and their properties are analyzed. Unlike linear filters which give good noise attenuation, but tend to smear edges and attenuate fine lines, these class of hybrid filters performs much better in preserving fine details. Also the ordering portion of the filters are independent of the size of the windows chosen.

**Algorithm A: Unidirectional FMH filters**

Assume a window size of 5x5, as is assumed in [NHN, 1987]. The algorithm of the basic unidirectional FMH filter is as follows:

$y(m,n) = \text{median}\{y_E(m,n), y_W(m,n), x(m,n)\}$

where the signals $y_i(m,n)$ are the outputs of FIR linear subfilters $h_i$. Index i indicates the direction where the filter $h_i$ is operating measured from the central input sample x(m,n). See Figure 2 for the interpretation of the subscripts indicating directions. It is possible to construct different unidirectional FMH filters which will have different noise attenuation properties. The masks for the unidirectional FIR linear FMH filters 1LH- is shown in Fig. 3(a) and its rotated versions R1LH-' R1LH-, and R1LH-" are shown in Figs. 3(b)-3(d) respectively. The number refer to the number in the median tree, which will be discussed later in the multilevel FMH filters. The dash stands for unidirectional operation, while the + stands for bidirectional operation as indicated below. As in [NHN, 1987], we assume the linear FIR subfilters to be of the averaging type.

Consider the 1LH- FMH filter. First we shall discuss how to compute both linear subfilters in one image processing pass.

Let P be the input image of size N × M. Let P1 be the result of translating P one pixel to the left, i.e. in the -x direction. Then P1[i,j] = P[i+1,j] for 1 <= i <= N-1, and for all j. If we sum P and P1 and divide the result by 2 (using a shift) pixelwise, and call the result image P2. Then P2[i,j] = average {P[i,j], P[i+1,j]}. Now notice that P2[i+3,j] = average { P[i+3,j], P[i+4,j] } which is the second linear subfilter for 1LH-. Hence the median of P[i,j], P2[i+2,j] and P2[i-1,j] for 3 ≤ i ≤ N-2 and for all j gives the value of pixel [i,j] of the result of the 1LH- FMH filter.

Let the original image P be stored in two image buffers, say A and B. Most pipeline image processor allows an image to be translated as it streams through the pipe from the buffer. Translating P one pixel in the -x direction as it leaves B allows P2 to be computed in one image processor pass. Let P2 be stored in image buffers, say C and D. Since both FIR linear filters can be recovered from P2 by translation, we can compute the two FIR subfilters in one image processor pass.

Define Tr[A, (p,q)] to be the resulting image of translating the image A p pixels in the x direction, and q pixels in the y direction. When p (or q) is negative, it indicates that the translation is in -x direction (-y direction).

Let A = P
Let B = Tr[P2,(i+2,0)]
Let C = Tr[P2,(i-1,0)]

Compute the median image of A, B and C using the following algorithm.

**Algorithm A1.**

Given input images A, B and C, the following image processing passes will be taken:

1. $\max(A, B) = X_1$
2. $\min(A, B) = X_2$
3. $\max(X_2, C) = X_3$
4. $\min(X_3, X_1) = X_4$

The output image is $X_4$

Then $X_4$ is the 1LH- filtered image. Note that A, B and C are used to denote images only. They do not necessarily need to be computed and stored.

**Theorem 1**

Algorithm A1 is correct

**Proof**

We need to prove that for each pixel [i,j] of $X_4$, it is the median of pixels [i,j] of A, B and C. We omit the details of the proof which can be found in [F, 1988].

□

The R1LH-' FMH filter can be computed by

(1) Translate P one pixel in the -y direction. let the result image be called Q1.

(2) Compute the average of P and Q1. Let the resulting image be called Q2.

(3) Compute FMH filter R1LH-' using Algorithm A1 where
A = P
B = Tr[Q2, (0,j+2)]
C = Tr[Q2, (0,j-1)]

The FMH filter R1LH-" corresponding to the two linear FIR subfilters from Fig. 2d can be computed as follows.

(1) Translate P one pixel in the +y direction and one pixel in the +x direction. Let the result image be called R1.

(2) Compute the average of P and R1. Let the resulting image be called R2.

(3) Compute FMH filter R1LH-" using Algorithm A1, where
A = P
B = Tr[R2, (-2,-2)]
C = Tr[R2, (1,1)]

**Algorithm B: Bidirectional FMH Filters**

The bidirectional masks for the 5x5 window are illustrated in Fig. 4(a) and (b). The 1LH+ FMH filter can be computed as follows. Let P be the original image.

(1) Translate P one pixel in the -x direction. Let the resulting image be called P1.

(2) Compute average of P and P1. Let the result be P2.

(3) Translate P one pixel in the -y direction. Let the result image be P3.

(4) Compute average of P and P3. Let the result be P4.

(5) Let A = P
Let B = Tr[P2, (i+2,0)]
Let C = Tr[P2, (i-1,0)]
Let D = Tr[P4, (0,j+2)]
Let E = Tr[P4, (0,j-1)]

Then the median of A[i,j], B[i,j], C[i,j], D[i,j] and E[i,j] for $3 \le i \le N-2$, and for $3 \le j \le M-2$ gives the value of the pixel [i,j] for the 1LH+ bidirectional FMH filter. This can be computed using Algorithm B1 below.

**Algorithm B1:**

Given input images denoted by A,B,C,D and E, the following image processing passes will be taken:

1) max (A,B) = $X_1$
2) min (A,B) = $X_2$
3) max ($X_2$, C) = $X_3$
4) min ($X_1$, $X_3$) = $X_4$
5) max ($X_1$, C) = $X_5$
6) min ($X_2$, C) = $X_6$
7) max ($X_4$, D) = $X_7$
8) min ($X_4$, D) = $X_8$
9) max ($X_8$, E) = $X_9$
10) min ($X_7$, $X_9$) = $X_{10}$
11) max ($X_{10}$, $X_5$) = $X_{11}$
12) min ($X_{10}$, $X_5$) = $X_{12}$
13) max ($X_{12}$, $X_6$) = $X_{13}$
14) min($X_{11}$, $X_{13}$) = $X_{14}$

The output image is $X_{14}$.

The correctness of Algorithm B1 will be discussed later in the paper.

The R1LH+ bidirectional FMH filter can be computed in similar fashion by defining the input to Algorithm B1, i.e. A,B,C,D and E accordingly. The details are as follows:

(1) Translate P one pixel in the +x direction and one pixel in the +y direction. Let the resulting image be called P1.
(2) Compute average of P and P1. Let the result be P2.
(3) Translate P one pixel in the +y direction and one pixel in the -x direction. Let the result image be P3.
(4) Compute average of P and P3. Let the result be P4.
(5) Let A = P
Let B = Tr[P2, (-2,-2)]
Let C = Tr[P2, (1,1)]
Let D = Tr[P4, (2,-2)]
Let E = Tr[P4, (-1,1)]
(6) Compute the median of A, B, C, D and E using Algorithm B1. The output of Algorithm B1 gives the value of the the 1LH+ bidirectional FMH filter applied to P.

In [NHN,1987], multilevel unidirectional and bidirectional FMH filters are introduced. The i-th level FMH filters are defined in terms of the i-1 level.

The algorithms for multilevel FMH filters are as follows.

**Algorithm C1: for the two-level unidirectional FMH filter**

(1) Compute 1LH-, R1LH-' FMH filters using Algorithm A
(2) Apply Algorithm A1 where
A = P
B = 1LH-
C = R1LH-'
The output of Algorithm A1 = 2LH- FMH filter

**Algorithm C2: for the rotated two-level unidirectional FMH filter**

(1) Compute R1LH-, R1LH-" FMH filters using Algorithm A
(2) Apply Algorithm A1 where
A = P
B = R1LH-
C = R1LH-"
The output of Algorithm A1 = R2LH- FMH filter

**Algorithm C3: for the three-level unidirectional FMH filter**

(1) Compute 2LH-, R2LH- FMH filters using Algorithm C1 and Algorithm C2 above.
(2) Apply Algorithm A1 where
A = P
B = 2LH-
C = R2LH-
The output of Algorithm A1 = 3LH- FMH filter

The bidirectional multilevel FMH filters can be computed using the algorithms below

**Algorithm C4: the two-level bidirectional FMH filter**

(1) Compute 1LH+, R1LH+ FMH filters using Algorithm B
(2) Apply Algorithm A1 where
A = P
B = 1LH+
C = R1LH+
The output of Algorithm A1 = 2LH+ FMH filter

**Algorithm C5: the rotated two-level bidirectional FMH filter**

(1) Compute R1LH+', R1LH+" FMH filters which are the outputs of five-point median operations over the pixels shown in Figs. 4(a) and 4(b). They are computed as follows.
i) Let A = P
Let B = Tr[P, (2,1)]
Let C = Tr[P, (2,-1)]
Let D = Tr[P, (-2,1)]
Let E = Tr[P, (-2,-1)]
Compute the median of A, B, C, D and E using Algorithm B1. The output of Algorithm B1 gives the R1LH+' bidirectional FMH filter.
ii) Let A = P
Let B = Tr[P, (1,2)]
Let C = Tr[P, (1,-2)]
Let D = Tr[P, (-1,2)]
Let E = Tr[P, (-1,-2)]
Compute the median of A, B, C, D and E using Algorithm B1. The output of Algorithm B1 gives the R1LH+" bidirectional FMH filter.

(2) Apply Algorithm A1 where
A = P
B = R1LH+'
C = R1LH+"
The output of Algorithm A1 = R2LH+ FMH filter

**Algorithm C6: the three-level bidirectional FMH filter**

(1) Compute 2LH+, R2LH+ FMH filters using Algorithm C4 and Algorithm C5.

(2) Apply Algorithm A1 where
A = P
B = 2LH+
C = R2LH+
The output of Algorithm A1 = 3LH+ FMH filter

## III. Using two comparators in parallel

The above discussion presents the algorithms in terms of the basic comparator operators. However, some of these operations can be computed in parallel if multiple processing units are available. In fact, many pipeline architectures are designed to consist of a number of parallel inputs into the image processor. For example, the DeAnza IP8500 has 8 parallel inputs with 4 sets of ALU and two comparators. Using two comparators, the algorithms corresponding to Algorithms A1 and B1 are presented below as Algorithms D1 and E1 respectively.

**Algorithm D1: For Unidirectional FMH filters**

This algorithm will be used in place of Algorithm A1 given above. Let A, B, C be three input images. Perform the following three image processing passes.

| | | | |
|---|---|---|---|
| 1. | Compute | max(A,B) | = E |
| | Compute | min(A,B) | = F |
| 2. | Compute | max(F,C) | = G |
| | Compute | max(E,C) | = H |
| 3. | Compute | min(G,E) | = I |
| | Compute | min(F,C) | = J |

Notice that we now have also computed:
H = MAX(A,B,C)
J = MIN(A,B,C)
where MAX(S) [i,j] = maximum{ P[i,j] | P in S} for set of images S
and MIN(S) [i,j] = maximum{ P[i,j] | P in S} for set of images S.

For bidirectional FMH filters, the algorithm is presented below.

**Algorithm E1: For bidirectional FMH filters**

This algorithm will be used in place of Algorithm B1 given above. Let A, B, C, D and E denote the five input images. The following image processing passes will be taken:

(1) Apply Algorithm D to (A,B,C)

let the resulting images be $(H_1, I_1, J_1)$

(2) Apply Algorithm D to ($I_1$,D,E)

let the resulting images be $(H_2, I_2, J_2)$

(3) Apply Algorithm D to $(H_1, J_1, I_2)$

let the resulting images be $(H_3, I_3, J_3)$

then

$I_3$ is the resulting bidirectional FMH filter.

**Theorem 2**

Algorithm E1 is correct.

**Proof**

The details of the proof can be found in [Fong,1988].

□

Now we can return to prove the correctness of Algorithm B1, which essentially follows from Theorems 1 and 2. Again we refer to [Fong, 1988] for details.

## IV. Analysis of the Algorithms

We shall derive the complexity of the algorithms in terms of the number of passes through the pipeline image processing system. In the DeAnza IP8500, the pipeline processor can perform comparisons at video refresh rate, i.e. 30 frames per second.

We assume a pipeline architecture with the minimum required resources, i.e. one hardware comparator. We also assume that an image can be translated in the x or in the y directions or in both directions as it leaves a refresh memory and streams through the pipe. This capability is available on most pipeline image processing systems by the use of look-up tables on each refresh memory board. In the following, we assume a $5 \times 5$ window.

For the unidirectional FMH filters, it takes one pass to compute the two subfilters, and four passes for Algorithm A1, for a total of 5 passes.

For the bidirectional FMH filters, it takes two passes to compute the four FIR subfilters, and fourteen passes for Algorithm B1, for a total passes of 16 passes.

For the multilevel FMH filters, both the two-level and the rotated two-level unidirectional FMH filter require 10 passes for the two first level FMH filters, plus 4 more for another application of Algorithm A1 for a total of 14 passes. The three-level unidirectional FMH filter requires 2 times 14 passes plus another application of Algorithm A1, for a total of 32 passes.

In the two-level bidirectional case, since each first-level bidirectional filter takes 16 passes, with another application of Algorithm A1, it totals 36 passes. The rotated two-level bidirectional FMH filter requires 2 applications of Algorithm B1 and one application of Algorithm A1, for a total of 32 passes. The three-level bidirectional FMH filter requires 36 + 32 and another application of Algorithm A1, using 4 passes, for a total of 72 passes.

For the general case with a window size of $n \times n$, in the unidirectional FMH case, the two linear subfilters can be computed using at most (n-1)/2 - 1 passes. This number can be improved to O(log n) passes rather than O(n) passes, while the number of passes to compute the median remains to be 4. In

the bidirectional FMH case, the four linear subfilters can be computed using at most 2 times (n-1)/2 -1, i.e. n-3 passes. Again, this can be improved to O(log n) passes. The number of passes to compute the median remains at 14.

We shall also derive the number of passes using the proposed algorithms for two hardware comparators. For the unidirectional FMH filters, it takes one pass to compute the two subfilters, and three passes for Algorithm D1, for a total of 4 passes.

For the bidirectional FMH filters, it takes two passes to compute the four FIR subfilters, and 9 passes for Algorithm E1, for a total passes of 11 passes.

For the multilevel FMH filters, both the two-level and the rotated two-level unidirectional FMH filter require 8 passes for the two first level FMH filters, plus 3 more for another application of Algorithm D1 for a total of 11 passes. The three-level unidirectional FMH filter requires 2 times 9 passes plus another application of Algorithm D1, for a total of 21 passes.

In the two-level bidirectional case, since each first-level bidirectional filter takes 11 passes, with another application of Algorithm D1, it totals 25 passes. The rotated two-level bidirectional FMH filter requires 2 applications of Algorithm E1 and one application of Algorithm D1, for a total of 21 passes. The three-level bidirectional FMH filter requires 25 +21 and another application of Algorithm D1, using 3 passes for a total of 49 passes.

The response of various FMH filters to test images and noisy images have been analyzed in [NHN, 1987]. The authors have concluded that one-level bidirectional FIR-median hybrid filter 1LH+ is best for images which contain mainly horizontal and vertical edges. If the image has randomly oriented edges, the best filtering result is achieved with the two-level bidirectional FIR-median hybrid filter 2LH+. Note that assuming two comparators and 30 frames per second rate, it takes less than half a second to compute 1LH+ and less than a second to compute 2LH+.

In the above analysis, we have assumed enough refresh memories so that no intermediate result has to be saved and reloaded from disk. Because of the amount of data involved in image data, any such transfer would be very time consuming. In Algorithm A1 and B1, a minimum of 4 refresh memories are necessary without disk transfer. In algorithms D1 and E1, 6 refresh memories are necessary to avoid disk transfer. Because of the parallel inputs usually designed for a pipeline processor, four refresh memories is typically the minimum required configuration in order to make effective use of the hardware. If only this minimum is available, it may be more advantageous to use algorithms using one comparator to eliminate the necessity for transfers.

Note that we cannot simply use an algorithm for finding the median of 5 elements by using comparisons as in sequential machines. This is because in sequential machine, a comparison of say a < b results in determining whether a or b is larger. This information can then be used to guide further testing. In the pipeline architecture, we do not have random access to each pixel in the memory, nor do we have the hardware logic to perform branching pixelwise as can be done in sequential machine. Note that using our algorithms, the final image give us the value of the median of the input images pixelwise. We do not know for each pixel which input image it comes from. What we desire is to apply the same operations to the entire images using only one frame time, rather than different operations on each pixel as with sequential machines. In fact, the correctness of the algorithms need to be proven.

## V. Conclusion

In this paper, we have presented a class of algorithms for fast computation for a class of detail preserving non-linear filters known as FIR hybrid median filters, using pipeline architectures with hardware comparators, which are available on most commercial pipeline image processing systems. The classes of filters considered are the unidirectional, the bidirectional and the multilevel FMH filters. Algorithms for hardware with either one or two comparators have been proposed. For proofs of correctness of the algorithms, refer to [Fong, 1988]. The analysis of the algorithms in terms of the number of pipeline passes and refresh memories required are also derived. In order to avoid disk transfer of images, one has to program the algorithms with great care and in the case where refresh memories is the limiting hardware resource, an algorithm which need more number of pipeline passes may be more appropriate.

Fast computation of these image enhancement filters using parallel architecture allow their use in realtime applications such as manufacture visual inspection. New area of their use may also be more easily explored in applications where speed is an important requirement and computationally intensive operations had not been previously viable.

## REFERENCES


[AAW, 1981] E. Ataman, V.K. Aatre, and K.M. Wong, "Some statistical properties of median filters", IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-29, pp. 1073-1075, Oct. 1981.

[AC, 1984] G.R. Arce and R.J. Crinon, "Median filters: Analysis for 2-dimensional recursively filtered signals", in Proc. IEEE ICASSP-84, San Diego, CA, Mar. 1984, pp. 20.11.1-20.11.4.

[AG, 1982] G.R. Arce and N.C. Gallagher, "State description for root- signal set of median filters", IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-30, pp. 894-902, Dec. 1982.

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

[DF, 1988] I. Dinstein and A.C. Fong (Lochovsky), "Computing Local Minima and Maxima of Digital Images in Pipeline Image processing Systems equipped with Hardware Comparators", Proceedings of the IEEE, to appear.

[DR, 1978] L.S. Davis and A. Rosenfeld, "Noise cleaning by iterated local averaging", IEEE Trans. Systems, Man and Cybernetics, vol. SMC-8, pp.705-710, Sept., 1978.

[F, 1988] Amelia C. Fong, "Fast Computation of a Class of Non-linear Detail Preserving Filters using Pipeline Architectures", Technical Report CIS88D2, Dept. of Computing & Information Science, University of Guelph, Mar., 1988.

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

[NHN, 1987] A. Nieminen, P. Heinonen and Y. Neuvo, "A New Class of Detail-Preserving Filters for Image Processing", in IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. PAMI-9, Jan. 1987, pp. 74-90.

[NG, 1983] T.A. Nodes and N.C. Gallagher, Jr., "Two-dimensional root structures and convergence properties of the separable median filter", IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-31, pp.1350-1365, Dec. 1983.

[NG, 1984] T.A. Nodes and N.C. Gallagher, Jr., "The output distribution of median type filters", IEEE Trans. Commun., vol. COM-32, pp. 532-541, May 1984.

[RNR, 1984] E.R. Ritenour, T.R. Nelson, and U. Raff, "Applications of the median filter to digital radiographic images", in Proc. IEEE ICASSP-84, San Diego, CA, Mar. 1984, pp. 23.1.1-23.1.4.

[S, 1988] J.L.C. Sanz, "A new method for computing polygonal marks in image processing pipeline architectures", Pattern Recognition, to appear.

[SD, 1987] Jorge L.C. Sanz and Its'hak Dinstein, "Projection-based Geometrical Feature Extraction for Computer Vision Algorithms: Algorithms in Pipeline Architectures", In IEEE Transaction on Pattern Analysis and Machine Intelligence, Jan. 1987, pp. 160-168.

[SDP, 1988] J.L.C. Sanz, I. Dinstein and D. Petkovic, "A new procedure for computing multi-colored polygonal masks in image processing pipeline architectures and its application to automatic visual inspection", to appear, CACM.

[SF, 1985] A. Stein and T.J. Fowlow, "The use of median filters for edge detection in noisy signals", in Proc. IEEE ISCAS-85, Kyoto, Japan, 1985, pp. 1331-1334.

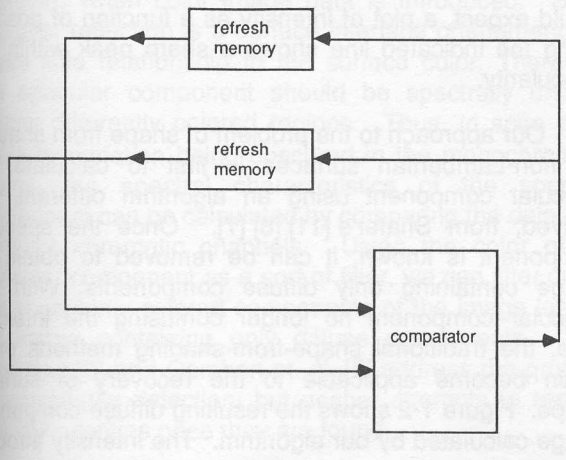


Fig. 1. Schematic Diagram for Pipeline Architectures.

$$h_N$$

$$h_W \quad x(m,n) \quad h_E$$

$$h_S$$

Fig. 2. Directions of FIR subfilters

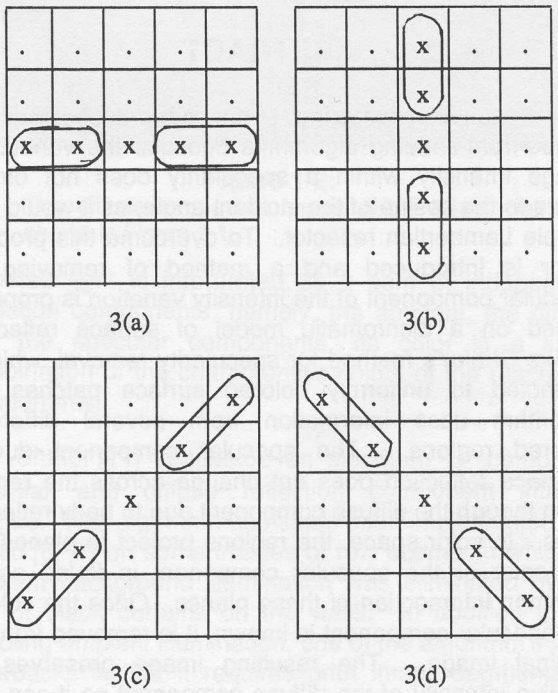


Fig. 3. Masks for unidirectional FMH filters
(a) 1LH- (b) R1LH- (c) R1LH-' (d) R1LH-"

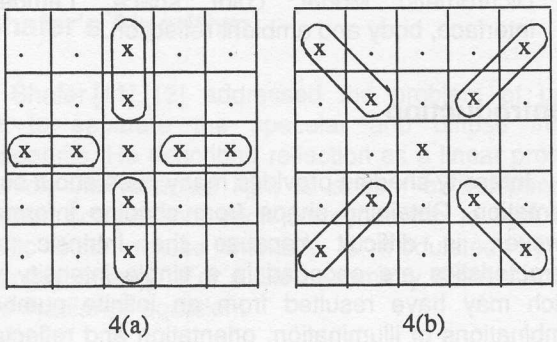


Fig. 4. Masks for bidirectional FMH filters
(a) 1LH+ (b) R1LH+