Comparison study of nonlinear filters in image processing applications

Yu-Shan Fong, MEMBER SPIE
Carlos A. Pomalaza-Rádez
Xiao-Hua Wang
Clarkson University
Department of Electrical and Computer Engineering
Potsdam, New York 13676

Abstract. Nonlinear filters are used in many applications, including speech and image processing, owing to their ability to suppress noise and preserve signal features such as edges. This study presents a performance evaluation of nonlinear filters derived from the robust point estimation theory. The first part of the work is a classification of various approaches to nonlinear filtering into three types of estimators according to the process of the filter. The second part is a computer implementation and evaluation of all of the filters discussed. Finally, a summary of experimental results is presented.

Subject terms: image processing; nonlinear filters; detail-preserving filters; robust point estimators; location parameters.

Optical Engineering 28(7), 749-760 (July 1989).

CONTENTS

1. Introduction
2. Classification of nonlinear filters
   2.1. Terminology and notation
   2.2. L-type filters
      2.2.1. α-trimmed mean filter
      2.2.2. Modified-trimmed mean (MTM) filter
      2.2.3. Double-window MTM filter
      2.2.4. K-nearest-neighbor (K-NN) filter
      2.2.5. Modified K-NN filter
   2.3. R-type filters
      2.3.1. Wilcoxon filter
      2.3.2. Limited degree Wilcoxon filter
      2.3.3. Finite-impulse-response median hybrid filter
   2.4. M-type filters
      2.4.1. Standard-type M-filter
      2.4.2. Adaptive mean filter
      2.4.3. Adaptive median filter
   2.5. Median class filters
      2.5.1. Conventional median filter
      2.5.2. Separate median filter
      2.5.3. Max/median filter
3. Comparison study
   3.1. Structure of test images
   3.2. Experimental study
   3.3. Statistical analysis by simulation
      3.3.1. Mean square error
      3.3.2. Subregion analysis
         3.3.2.1. Step response
         3.3.2.2. Line response
4. Conclusion
5. References

1. INTRODUCTION

Linear and nonlinear filters are used extensively in signal processing applications to remove noise. It has been shown by many experimental studies that although linear filters possess good noise attenuation capabilities, they smear the edges and attenuate thin lines present in the original signal because of the linear averaging operation that they perform. On the other hand, nonlinear filtering is a well-known noise filtering and edge-preserving method. This class of filters has become popular in digital speech and image processing and has achieved some interesting results in many image processing applications.

Tukey1 first used the median filter for nonlinear smoothing of data. This filter became attractive because it is easy to implement and can reduce quite effectively the impulsive noise component. Median filtering is a local filtering technique in which each pixel is replaced by a value obtained through a "median" operation performed within a window. The term window refers to a neighborhood of the pixel, centered at that pixel. As an example, Fig. 1 shows the performance of a median filter with a square window of size 3×3 for the removal of impulse noise.

Median filters have been applied to several areas of digital signal processing, including image processing and speech processing.2,3 One main feature of the median filter is that it eliminates the impulse that has a duration of less than half the window size. Since the degree of smoothing of the median filter can be influenced only by the processing window size, it does not generally allow the user sufficient control over its characteristics. Furthermore, it does not have the averaging operation that is particularly appropriate in reducing additive Gaussian noise components. Several techniques have been introduced in an effort to overcome these limitations. One is to combine linear and nonlinear operations. The combination allows a degree of control over the relative influence of these operations. Bednar and Watt4 proposed such a new algorithm for applying median filtering. It is called the α-trimmed mean (α-TM) filter. Suppose that X is a finite set of N numbers. The α-TM of X is obtained by sorting X into rank order, removing
a fixed fraction $\alpha$ ($0 \leq \alpha \leq 0.5$) from the high and low ends of the sorted set, and computing the average of the remaining values. The procedures they developed have opened up a new class of nonlinear filters based on indexing schemes. Lee and Kassam introduced two useful new $\alpha$-TM filters, the modified-trimmed mean (MTM) filter and the double-window modified-trimmed mean (DWMTM) filter. These two filters have achieved more smoothing without severe loss of narrow pulse in an image. The MTM filter does not have a fixed fraction $\alpha$. It is changed every time by comparison with a preselected threshold. It has a nonsymmetric data-dependent smoothing property. The DWMTM filter has the same characteristics as the MTM filter but preserves the narrow lines better by having two window processings.

Davis and Rosenfeld developed a class of nonlinear filters called the K-nearest-neighbor (K-NN) filters. This was later extended to the modified K-nearest-neighbor (MK-NN) filter by Peterson and Kassam. The K-NN filter is similar to the $\alpha$-TM filter in that it takes the average of a fixed number of selected values for each window processing, so they have a similar performance in nonimpulsive noise component suppression, but the K-NN filter can be more effective than the $\alpha$-TM filter in edge preservation for properly chosen K. This is because it always averages K consecutive pixel values from among the ordered set, but the location of these values in the ordered set is not fixed. In other words, like the MTM filter, K-NN filters have a nonsymmetric data-dependent smoothing property.

The Wilcoxon filter was introduced by Crinon, and an extension of the Wilcoxon filter was proposed by Song and Kassam. This new kind of rank-order filter scheme also has both averaging and nonlinear characteristics. Heinonen and Neuvo introduced a new class of median filter called smoothed median (SM) filters. The input signal X is filtered with M linear phase finite-impulse-response (FIR) filters, and the output of the filter is the median of the outputs of the FIR filters. Based on this, Nieminen, Heinonen, and Neuvo developed a class of filters called FIR-median hybrid (FMH) filters. The concept of multilevel median operation is specifically pointed out here. This kind of filter performs better in preserving details in the image.

Since the problem of removing the additive noise produced by the imaging system without blurring the fine details of the image arises, Pomalaza-Ráez and McGillem have proposed an adaptive nonlinear edge-preserving filter whose main feature is that the window size adjusts automatically, depending on the nature of the signal itself.

Each of these filters shows optimal smoothing efficiency for only a specific type of noise of a specific type of image. To further apply these filters in image processing properly, it is important to review and analyze all of these filters and to objectively evaluate their performance in both noise-removing and edge-preserving abilities.

This study presents a performance evaluation of nonlinear filters derived from the theory of robust estimation of location parameters. There are three classes of estimation that can be distinguished, namely the L-estimator, based on linear combination of order statistics; the R-estimator, derived from rank tests; and the M-estimator, or the maximum likelihood estimator. Section 2 of this paper describes the first part of the work carried out in this study, which classifies various approaches of nonlinear filtering into these three types of estimators according to the behavior of the filter. For comparison, we also implemented the median filters. Since the median filter is a special case of L-, R- and M-filters, it is also regarded as a robust estimation technique.

Section 3 presents the second part of the work in this study, which implements the filters discussed in Sec. 2 on a personal computer and presents the experimental results illustrating their performance. The synthesized test images are added with Gaussian noise and are filtered. The resulting image is analyzed using mean square error (MSE) and local statistics, and the performance of the filters is compared.

2. CLASSIFICATION OF NONLINEAR FILTERS

2.1. Terminology and notation

Let us first define some useful terminology and notation. Consider an input sequence $X(i)$. The input is composed of a signal component degraded by additive noise. The filter output sequence $Y$ at index $k$ is denoted as $Y_k$ and is computed as a function of the input value from within a sliding window of size $W$. Hence, we define the windowed input sequence at index $k$ as

$$\bar{X}_k = \begin{bmatrix} x_{1,1} & \cdots & x_{1,(2m+1)} \\ \vdots & & \vdots \\ x_{(2m+1),1} & \cdots & x_{(2m+1),(2m+1)} \end{bmatrix}$$

where $W = 2N + 1 = (2m + 1)^2$ is the window size and $m, N$ are all integers; $x_{k,(N+1)}$ is the central input sample.

The rank of the $i$th input value from the window at index $k$ is given by $R_k(i)$, and the sequence of order statistics is denoted by $Z_k$. Then

$$\bar{Z}_k = (Z_k(1), \ldots, Z_k(W))^T$$

where $Z_k(1) \leq Z_k(2) \leq Z_k(W)$. Later, we will always consider time invariant processing of data derived from a local window centered about the current index $k$. The original data sequence $\bar{X}_k$ may be ranked ordered to yield ranks $\bar{R}_k$ and order statistics $\bar{Z}_k$.
2.2. L-type filters

The output $Y_k$ of an L-filter may be expressed as a fixed linear combination of the order statistics:

$$Y_k = \sum_{i=1}^{w} a(i)Z_k(i),$$  \hspace{1cm} (4)

where $Z_k(i)$ is the $i$th sample of order statistics among the $2N+1$ samples inside the window centered at $k$ and $a(i)$ is the data-dependent coefficient, which is a set of constant weights.

The L-filter structure, based upon the statistical L-estimator of location, treats each input as having the same mean. The input is assumed to be locally constant, and the temporal ordering of the input is abandoned. By selecting different L-filter data-dependent coefficients $a(i)$, different structures of the L-type filter are realized.

2.2.1. $\alpha$-TM filter

The output of this process at index $k$ is

$$Y_k = \frac{1}{W-2\lceil \alpha W \rceil} \sum_{i=\lceil \alpha W \rceil+1}^{W} Z_k(i),$$  \hspace{1cm} (5)

where $\lceil \cdot \rceil$ is the greatest integer function and $Z_k(i)$ represents the $i$th data item in the sorted sample within the window. The output of this filter is the trimmed mean value. The trimmed mean is so called because, rather than averaging the entire data set, a few data points are removed (trimmed) and the remainder are averaged. The number of data values that are removed is a function of the number of data values $N$, the trimming parameter $\alpha$, which assumes values between 0 and 0.5. In this filter the points that are removed are the most extreme values, both low and high, with an equal number of points dropped at each end.

2.2.2. MTM filter

The output of this process at index $k$ is

$$Y_k = \frac{\sum_{i=1}^{2N+1} a(i)Z_k(i)}{\sum_{i=1}^{2N+1} a(i)},$$  \hspace{1cm} (6)

where $a(i)$ is a weight function in the form

$$a(Z_k(i),Z_k(N+1)) = \begin{cases} 1, & \text{if } |Z_k(i) - Z_k(N+1)| < \varepsilon, \\ 0, & \text{otherwise}. \end{cases}$$  \hspace{1cm} (7)

The number of values used in averaging is not fixed in MTM processing as it is in $\alpha$-TM processing, so MTM processing can be thought of as a modification of the $\alpha$-TM filter.

2.2.3. DWMTM filter

This process uses two windows $W_1 = 2N+1$ and $W_2 = 2L+1$, where $L > N$. The output of this process is

$$Y_k = \frac{\sum_{i=1}^{2L+1} a(i)Z_k(i)}{\sum_{i=1}^{2L+1} a(i)},$$  \hspace{1cm} (8)

where $Z_k(i)$ represents the order statistics of the large window $W_2$, and $a(i)$ is a weight function in the form shown in Eq. (7).

The median value $Z_k(N+1)$ is the median value of pixels within the small window $W_1$ centered on any pixel location $k$. The large window permits more noise suppression than is possible with a smaller window size for short pulse retention.

2.2.4. K-NN filter

The output of this process is

$$Y_k = \frac{\sum_{i=1}^{2N+1} a(i)x_k(i)}{\sum_{i=1}^{2N+1} a(i)},$$  \hspace{1cm} (9)

where $a(i)$ is in the form

$$a(i) = \begin{cases} 1, & \text{if } x_k(i) \text{ is one of the } K \text{ closest values to } x_k(N+1), \\ 0, & \text{otherwise}. \end{cases}$$  \hspace{1cm} (10)

Two versions of the K-NN filter may be defined. The first version, K-NN$_1$, takes the average of a fixed number $K$ of values in the window closest to the subject pixel, including the pixel itself, where $K$ is a preselected integer less than or equal to the window size $2N+1$. The second version, K-NN$_2$, takes the same average except that the subject pixel is excluded from the averaging. In other words, the window $W$ does not include the subject pixel.

2.2.5. MK-NN filter

The output of this process at index $k$ is as shown in Eq. (6), but here $a(i)$ is in the form

$$a(x_k(i),Z_k(N+1)) = \begin{cases} 1, & \text{if } x_k(i) \text{ is one of the } K \text{ values closest to the value of the median pixel }, \\ 0, & \text{otherwise}. \end{cases}$$  \hspace{1cm} (11)

This processing leads to a filter that averages $K$ pixels whose gray levels are closest to the median value among the gray levels in any window ($K \leq 2N + 1$).

2.3. R-type filters

This class of filter has its basis in the rank estimate of statistical theory.\textsuperscript{9} For an input sequence $\{X(i)\}$, the output $Y_k$ at index $k$ of window size $2N+1$ is the value of the rank estimate.

Therefore, the output of an R-filter is affected by the relative ranks of data and not by the actual values of the data. It may be expressed as

$$Y_k = \text{order statistics of } \{F[X(i)]\},$$  \hspace{1cm} (12)

where $F[X(i)]$ could be any linear function of the input data sequence $\{X(i)\}$. By use of different linear functions of the input data, different structures of R-filters can be formed.

2.3.1. Wilcoxon filter

The output of this process at index $k$ is

$$Y_k = \text{median of } \left\{ \frac{x_j(k) + x_\ell(k)}{2} : \text{for all possible } j \text{ and } \ell \right\},$$  \hspace{1cm} (13)

where $j$ and $\ell$ are from the same row or column in the window.
The subscripts 1, 2, . . ., M represent the number of subfilters, and \( y_j(t) \) is the output of each subfilter, where \( t = 2h \) is the number of pixels used by the subfilter; \( x_k(N + 1) \) is the central pixel. The masks covering the input pixels of the subfilters are located symmetrically with respect to the center of the window.

Two types of FMH filters can be constructed: the unidirectional and the bidirectional. In the unidirectional FMH filters the masks and the central input are on the same line, whereas in the bidirectional filters the mask of the subfilters is either in the horizontal and vertical direction or rotated 45°. More specifically,

\[
Y_k = \text{median of } [y_E(t), y_S(t), y_W(t), y_N(t), x_k(N + 1)] , \quad (16)
\]

where the subscripts E, S, W, N, NE, SE, SW, and NW represent the directions from the center pixel \( x_k(N + 1) \) and the \( y_j(t) \) are the outputs of the linear subfilters.

Examples of simple unidirectional FMH masks for a 5 X 5 data window with (a) \( h = 1 \), (b) \( h = 2 \), and (c) \( h = 3 \) are shown in Fig. 2.

The unidirectional FMH filter is also called a 2-D smoothed median filter. In the discussion related to the experiments, we refer to the unidirectional FMH filter as the SM filter.

2.4. M-type filters.

The output \( Y_k \) of an M-type filter is defined as a solution of the equation

\[
\sum_{i=1}^{n} \psi(x(i) - Y_k) = 0 , \quad (18)
\]
where $\psi$ is some odd, continuous, and sign-preserving function. Suppose there exists a set of observations $x(i) = \theta + W(i), i = 1, \ldots, n$, where $\theta$ is a location parameter and $\{W(i)\}^n_{i=1}$ is a sequence of i.i.d. zero-mean noise components. Then for some even function $p(x)$ that is nondecreasing in $|x|$, we can define an estimator $\hat{\theta}$ for $\theta$ that minimizes $\sum_{i=1}^{n} p(x(i) - \hat{\theta})$ or that equivalently satisfies

$$\sum_{i=1}^{n} \psi(x(i) - \hat{\theta}) = 0 \, ,$$

where $\psi(x) = dp(x)/dx$ is an odd function, i.e., $\psi(x) = -\psi(-x)$.

Such estimates are generalized forms of maximum-likelihood estimates, and the estimator is called the M-estimator. A filtering procedure that uses a running M-estimator will be called an M-type filter.

### 2.4.1. STM filter

According to the definition of the M-type filter, the standard-type M-filter has a $\psi(x)$ function for which

$$\psi(x) = \begin{cases} g(p) , & x > p , \\ g(x) , & |x| \leq p , \\ -g(p) , & x < -p , \end{cases}$$

where $g(x) = ax$, and $p$ is a positive constant.

Consider the behavior of the STM filter. The output at index $k$ is $Y_k$, and the $x$ are the samples inside the window. The STM filter replaces these samples outside a range $[\text{med}_k - q_1, \text{med}_k + q_2]$ with $\text{med}_k - q_1$ or $\text{med}_k + q_2$, depending on whether $x_k \leq \text{med}_k - q_1$ or $x_k \geq \text{med}_k + q_2$, respectively, where $q_1 = \text{med}_k - Y_k + p$, $q_2 = Y_k + p - \text{med}_k$ and $\text{med}_k$ denotes the median of samples in this window. Here, $q_1$ and $q_2$ also depend on the data. The final output of the STM processing is the average value over this new set of data samples. The functions of the L-type MTM filter and the STM filter are similar. They differ in that the STM filter only limits the influence of some data values, whereas the MTM filter may reject certain data values.

### 2.4.2. Adaptive mean filter

The output of this process is as shown in Eq. (9), and here

$$a(i) = \begin{cases} 1 , & \text{if} |x_k(N + 1) - x_k(i)| \leq C , \\ 0 , & \text{if} |x_k(N + 1) - x_k(i)| > C , \end{cases}$$

for $i = 1, 2, \ldots, 2N + 1$.

The value of $C$ in the equation depends on the variance of the contaminating noise. It can be chosen according to some optimization criterion that equals $3\sigma$ (standard deviation of noise). The concept behind this processing is to have a filter whose window dimensions adjust automatically. The window size depends on the nature of the signal being processed, whether it is an edge or a smooth region.

### 2.4.3. Adaptive median filter

This filter is similar to the adaptive mean filter. The output of an adaptive median filter is

$$Y_k = \text{median of } \{x_k(i) | x_k(i) \in S \} \, ,$$

where

$$x_k(i) \in S , \text{ if } |x_k(N + 1) - x_k(i)| \leq C ,$$

$$x_k(i) \notin S , \text{ if } |x_k(N + 1) - x_k(i)| > C \, ,$$

and $i = 1, 2, \ldots, 2N + 1$. The main difference here is that instead of taking the average of the new set of samples within the window, it takes the order statistics of the new set of samples.

### 2.5. Median class filters

It is well known that the median filter is a special case of L-, R-, and M-filters. In extending the one-dimensional median filter to two dimensions, different processes have been proposed. In our experiments, we implement three types of median filters. They are the conventional median filter, the separate median filter, and the max/median filter.

#### 2.5.1. Conventional median filter

This is the "generic" square-window median. The output is the median of all of the elements in the running window.

#### 2.5.2. Separate median filter

The output of the separate median filter is

$$Y_k = \text{median of } \{z_1, z_2, \ldots, z_{2m+1}\} \, ,$$

where

$$z_i = \text{median of } \{x_{i-1}, x_i, \ldots, x_{i+2m+1}\}$$

and $x_{i,j}$ denotes the element in the $(2m+1) \times (2m+1)$ window (see Sec. 2).  

#### 2.5.3. Max/median filter

The max/median filter was proposed by Arce and McLoughlin. Its output can be defined as

$$Y_{k,i,j} = \max(Z_i, Z_3, Z_4) \, ,$$

where

$$Z_i = \max \{x_{i,j-m}, \ldots, x_{i,j}, \ldots, x_{i,j+m}\} ,$$

$$Z_3 = \max \{x_{i-m,j}, \ldots, x_{i,j}, \ldots, x_{i+m,j}\} \, ,$$

and

$$Z_4 = \max \{x_{i-m,j-m}, \ldots, x_{i,j}, \ldots, x_{i+m,j+m}\} \, .$$

This is a two-pass process. In the first pass, the median of the points along each of the four lines through the center pixel (vertical, horizontal, and the two diagonals) is taken. In the second pass, the maximum of these medians is identified and regarded as the output.

### 3. COMPARISON STUDY

#### 3.1. Structure of test images

Two test images are used in this experimental study. Each consists of $128 \times 128$ pixels. To eliminate the effects of image
The criterion used to compare the performance of various filters is the MSE. We use the MSE of an image (noisy background is 20). This test image was used to examine the filter's ability to preserve thin lines and sharp corners.

3.2. Experimental study

The generated test images are added with zero-mean Gaussian random noise. After implementing all of the previously discussed filters on test images I and II, we compare the performance of the filters using empirical statistical analysis. Performance evaluation of the filter in noise reduction is carried out from the MSE viewpoint. The evaluation of detail-preserving capabilities of the filters is carried out using local statistics.

The 3 X 3 window is used in all of the filter implementations except for FMH filters (including the SM filter), for which a 5 X 5 window is used.

3.3. Statistical analysis by simulation

3.3.1. MSE

The criterion used to compare the performance of various types of filters is the MSE. We use the MSE of an image (noisy background is 20). This test image was used to examine the filter's ability to preserve thin lines and sharp corners.

Test image I, shown in Fig. 5 in a three-dimensional perspective view, consists of a solid square object and a solid triangular object on a darker background. The gray level of the triangle is 40, that of the square is 30, and that of the background is 20.

Test image II, shown in Fig. 6, contains several lines and patterns in a variety of directions with widths of either one or two pixels. The gray levels of these patterns are either 40 or 30, the gray level of the triangle is 40, that of the square is 30, and that of the background is 20. This test image was used to examine the filter's ability to preserve thin lines and sharp corners.

<table>
<thead>
<tr>
<th>TABLE I. Mean square error of filtered images.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td>----------------</td>
</tr>
<tr>
<td><strong>L-type filters</strong></td>
</tr>
<tr>
<td>$\alpha$-Trimmed Mean</td>
</tr>
<tr>
<td>($\alpha = 0.1$)</td>
</tr>
<tr>
<td>($\alpha = 0.2$)</td>
</tr>
<tr>
<td>($\alpha = 0.3$)</td>
</tr>
<tr>
<td>($\alpha = 0.4$)</td>
</tr>
<tr>
<td>Modified-Trimmed Mean</td>
</tr>
<tr>
<td>($q = 5$)</td>
</tr>
<tr>
<td>($q = 10$)</td>
</tr>
<tr>
<td>($q = 15$)</td>
</tr>
<tr>
<td>($q = 20$)</td>
</tr>
<tr>
<td><strong>K-nearest neighbor</strong></td>
</tr>
<tr>
<td>($K = 2$)</td>
</tr>
<tr>
<td>($K = 4$)</td>
</tr>
<tr>
<td><strong>Modified-KNN</strong></td>
</tr>
<tr>
<td>($K = 2$)</td>
</tr>
<tr>
<td>($K = 4$)</td>
</tr>
<tr>
<td><strong>Double window MTM</strong></td>
</tr>
<tr>
<td>($q = 5$)</td>
</tr>
<tr>
<td>($q = 10$)</td>
</tr>
<tr>
<td>($q = 15$)</td>
</tr>
<tr>
<td>($q = 20$)</td>
</tr>
<tr>
<td><strong>R-type filters</strong></td>
</tr>
<tr>
<td>Wilcoxon filter</td>
</tr>
<tr>
<td>LDW</td>
</tr>
<tr>
<td>($M=2$, $D=2$)</td>
</tr>
<tr>
<td>($M=3$, $D=3$)</td>
</tr>
<tr>
<td>Smooth-median</td>
</tr>
<tr>
<td>RFMH</td>
</tr>
<tr>
<td>MFMH</td>
</tr>
<tr>
<td><strong>M-type filters</strong></td>
</tr>
<tr>
<td>Adaptive Mean</td>
</tr>
<tr>
<td>($C = 27$)</td>
</tr>
<tr>
<td><strong>Median class filters</strong></td>
</tr>
<tr>
<td>Conv. Median</td>
</tr>
<tr>
<td>Separate Median</td>
</tr>
<tr>
<td>Max/Median</td>
</tr>
</tbody>
</table>
or filtered) as a measure of its deviation from the original
(noise free) image. Thus, the MSE for a noisy image is

\[
\text{MSE} = \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} (X_0(i,j) - X_n(i,j))^2 , \quad (28)
\]

where \(X_0(i,j)\) is the \(i\)th and \(j\)th pixels in the original image and
\(X_n(i,j)\) is the \(i\)th and \(j\)th pixels in the noisy image. We use the
same expression to calculate the MSE of the filtered image.
The only difference is that \(X_0(i,j)\) is changed to \(X_f(i,j)\) for
filtered images.

Table I provides the results of calculating the MSE of the
filtered image by various types of filters. In the table, the
subscripts I and II represent test images I and II, respectively.
R is the ratio of the MSE of the filtered image to that of
the noisy image. In other words, \(R = \text{MSE}_f / \text{MSE}_n\), where
the MSE of the noisy image is equal to 8.7023 for both test image I
and test image II.

### 3.3.2. Subregion analysis

To observe the edge-preserving ability of each filter, we calculate
the local mean, local variance, and rms error between the
filtered image and the original noise-free image in different
subregions.

By comparing the local mean of the input and output
distributions, we obtain information about how well the filter
can preserve details of the image. In other words, the discrepancy
between these two local means is an indication of the
degree of edge blurring of each filter. The local mean \(M_t\) is
defined as

\[
M_t = \frac{1}{W} \sum_{i=1}^{W} X(i) , \quad (29)
\]

where \(W\) is the subregion size.

Comparing the local variance of the filtered output distribution
with the local variance of the original noise-free image, the
diffusion of each filter at the edge or the discontinuity
points can be shown. It is represented by \(V_t\) and is defined as

\[
V_t = \frac{1}{F} \sum_{i=1}^{W} (S(i) - M_t)^2 . \quad (30)
\]

The value of the local rms error in each subregion is represented by \(\text{RMS}_t\), which is defined as

\[
\text{RMS}_t = \frac{1}{F} \sum_{i=1}^{W} (\theta[X(i)] - S(i))^2 , \quad (31)
\]

where \(\theta[\cdot]\) represents the filter operation, \(S(i)\) is the pixel
value in the noise-free image, and \(F\) is a normalizing factor.
This calculation gives information about both the edge-pre-
serving and the noise-cleaning ability. It shows the deviation of
the filtered image from the original image.

#### 3.3.2.1. Step response

To analyze the filter’s ability to preserve edges under noise, we
use test image I and select a row of 14 pixels that contains an
ideal step function. After the whole image is filtered, we slide a
1-D window across the step function, calculating the local
mean, local variance, and rms error at each window location.
Here, the window size \(W\) is 3.

**Local mean**

Figures 7(a) through 7(f) show the local mean of the filter
output sample on both sides of the edge for each filter.

**L-type filters:** All of the \(\alpha\)-TM class filters [Fig. 7(a)] do
acceptably well on the edge. They are mostly parallel to the
Fig. 8. Local variance of filtered step response: (a) α-TM, (b) K-NN, (c) Wilcoxon, (d) FMH, (e) M-type, and (f) median.

edge of the noise-free image, but when compared with the curve of the noise-free image on the smooth region, which has the value 20 before the edge and 40 after the edge, both α-TM and MTM filters have some distortion on those regions. The degree of distortion reflects the noise reduction ability on the smooth region of the filters. The K-NN class filters [Fig. 7(b)] have almost the same performance as α-TM class filters in terms of edge preserving, but the K-NN filters have more distortion on the smooth region.

R-type filters: The Wilcoxon filter [Fig. 7(e)] makes the step spread out to a ramp. It displays poor performance on the edge. The LDW filter improves the edge-preserving ability, but it has distortion on the two sides of the edge. The FMH class filters [Fig. 7(d)] do acceptably well too, except for the LFMH filter, which seems to display very bad performance. This is due to the orientation of the edge. For the particular edge we have chosen here, its performance is quite poor. This shows again that the FMH class filters are quite sensitive to orientation.

M-type filters: Although the adaptive mean filter [Fig. 7(e)] has much better performance on the edge than does the adaptive median filter, when compared with other filters it has more distortion on the two sides of the edge. That means its ability in noise reduction on the smooth region is not as good as that of other types of filters.

Median class filters: The conventional and the separate median filters [Fig. 7(f)] perform well on the edge. The degree of edge blurring is very small. The separate median filter also gives very good performance on the smooth region. The max/median filter has more distortion on the two sides of the edge.

Local variance

Figures 8(a) through 8(f) show the local variance of the filtered output for each filter. They are all compared with the local statistics of the original image. Basically, we are looking for the width of the local variance curve. This represents the degree of diffusion at the edge by each filter.

L-type filters: As seen in Fig. 8(a), the DWMTM filter gives a very good response, as we expected. At the edge points (pixels six and seven) it has almost the same width and the same amplitude as the original. Also, consider the two sides of the edge. Compared with the noise-free curve, it shows very good performance.

The K-NN filter [Fig. 8(b)] has high variance at the edge points. The MK-NN filter performs better. Also, the K-NN filter has more distortion on both sides of the edge than do the MK-NN filter and the DWMTM filter.

R-type filters: The LDW filter [Fig. 8(e)] has almost the same performance as the MK-NN filter, but the Wilcoxon filter has the worst performance. It spreads the width of the distribution curve by 25% on both sides and decreases the amplitude by 55%.

The LFMH filter [Fig. 8(d)] has a high degree of edge blurring; its width of the curve spreads by 25%. (This is owing to the sensitivity of orientation, as mentioned before.) All other FMH class filters show good concentration at the edge points (pixels six and seven), but the distortion on the two sides of the edge obviously is worse than for the LDW filter.

M-type filters: As discussed before, the adaptive mean filter [Fig. 8(e)] has very good performance. The width of the distribution is almost the same as that of the noise-free distribution. Also, on the two sides of the edge there is a very small degree of distortion. The adaptive median filter, however, has very poor edge-preserving ability. Although it has high variance, the width of the distribution is totally distorted.

Median class filters: Median class filters [Fig. 8(f)] give an average performance at this point. In general, they have good edge-preserving ability.
Local rms error

Figures 9(b) through 9(g) show the rms-error distributions for each filter. As a reference, we plot the local rms error of the noisy image in Fig. 9(a). From the result, we see that the DWMTM [Fig. 9(b)] has the best performance in this respect, and the adaptive median filter [Fig. 9(f)] has the worst. The LFMH filter [Fig. 9(e)] does not fare well either.

3.3.2.2. Line response

To analyze the filter's ability to preserve a narrow line under noise, we used test image II. We select a segment containing two thin lines; each line is one pixel wide and the total segment has 14 pixels. The procedure is the same as for the step response. Figures 10(a) through 10(f) show the local mean of the filtered output distribution for each filter, and Figs. 11(a) through 11(f) show the local variance of the filtered output for each filter.

Local mean and local variance

L-type filters: It is obvious that all of the \( \alpha \)-TM class filters have poor performance in this case. They spread the line and decrease the amplitude by almost 50%. Also, they cannot follow the jump, as can be seen in the local variance curves. Even the DWMTM filter has poor performance. The MK-NN filter does better in both the local mean and local variance sense.

R-type filters: Compared with the L-type filter, the Wilcoxon filter shows some improvement. It does not spread the line that much. There is an exciting result with FMH class filters. Except for the LFMH filter, which has totally failed the test (we mentioned the reason before), all other types of filters have good response to this one-pixel-wide line.

M-type filters: Figure 10(c) shows that the adaptive mean filter has the best performance in line response. Except for the distortion at the smooth region on the two sides of the line, it...
gives good response for both local mean and local variance distributions.

**Median class filters:** The conventional and separate median filters all failed the thin-line test. They spread the line and decrease the amplitude by 65%. The max/median filter, however, shows good performance. It has higher amplitude and does not spread the width. It has very good response for the local variance distribution. This shows that the max/median filter has good thin-line preserving ability, which is owing to the max operation.
COMPARISON STUDY OF NONLINEAR FILTERS IN IMAGE PROCESSING APPLICATIONS

Local rms-error

Figures 12(a) through 12(f) give the local rms error of the filtered output distributions for each filter. Among them, the MK-NN, the adaptive mean, and the MFMH filters have the best performances.

4. CONCLUSION

We have studied the relative performances of different types of nonlinear filters. The abilities of various filters are tested using synthetic images.

The goal of this study was to review and analyze different types of nonlinear filters and to objectively evaluate their performances. As a conclusion, we give an overview of the analysis in Table II.

From this study, the following can be observed:

The M-type adaptive mean filter provides better overall characteristics for filtering. With the optimum choice of the threshold \( C(C = 3\sigma) \), it performs best in edge and line preservation; also, it has good ability in noise reduction. The adaptive mean filter is very attractive for image processing applications.

The L-type filters have distinctive characteristics such as strong ability in Gaussian noise reduction and very good edge preservation. In terms of MSE, the L-type double window modified trimmed mean filter gives the best result \( (R = 0.14) \), but because of the double window method for each pixel processing, the computation time is almost doubled. Also, it must be noted that L-type filters have totally failed the thin-line and impulse edge-preserving tests.

The R-type filters were shown to preserve more subtle details than did the L-type filters. All of the analyzed FIR-median hybrid filters are able to preserve one-pixel-wide lines in some directions. In particular, the multilevel FMH filter has shown good performance in edge and line preservation.

<table>
<thead>
<tr>
<th>TABLE II. Overview of the performance of various filters.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filter Type</td>
</tr>
<tr>
<td>L-type Filters</td>
</tr>
<tr>
<td>α-TM filter</td>
</tr>
<tr>
<td>MTM filter</td>
</tr>
<tr>
<td>K-nearest neighbor</td>
</tr>
<tr>
<td>Modified-KNN</td>
</tr>
<tr>
<td>DM-MTM</td>
</tr>
<tr>
<td>R-type filters</td>
</tr>
<tr>
<td>Wilcoxon filter</td>
</tr>
<tr>
<td>LDWF filter</td>
</tr>
<tr>
<td>Smooth-median</td>
</tr>
<tr>
<td>LFMH filter</td>
</tr>
<tr>
<td>RFMH filter</td>
</tr>
<tr>
<td>MFMH filter</td>
</tr>
<tr>
<td>M-type filter</td>
</tr>
<tr>
<td>Adaptive Mean</td>
</tr>
<tr>
<td>Adaptive Median</td>
</tr>
<tr>
<td>Median Class filter</td>
</tr>
</tbody>
</table>

OPTICAL ENGINEERING / July 1989 / Vol. 28 No. 7 / 759
However, its ability in noise reduction is not as good as that of L-type filters.

Median filters have average performance. In particular, the overall performance of the max/median filter is not as good as that of the adaptive mean filter, but its performance as a thin-line preserver is quite acceptable.

5. REFERENCES