

# Nonlinear Smoothing Filters Based on Rank Estimates of Location

PRASHANT P. GANDHI, STUDENT MEMBER, IEEE, HICKHO SONG, MEMBER, IEEE, AND  
SALEEM A. KASSAM, SENIOR MEMBER, IEEE

**Abstract**—A class of nonlinear filters is introduced, which is based on the rank estimates ( $R$ -estimates) of location parameters in statistical theory. We show first how moving-window rank filters ( $R$ -filters) can be defined starting from rank estimates of location. These filters utilize the relative ranks of the observations in each window to produce an output value. A special class of rank filters produces outputs which are medians of selected pairwise averages of observations inside each window. The Wilcoxon filter is one simple example of such an  $R$ -filter. The concept of rank Winsorization allows a limiting of the influence of larger rank observations. We extend the idea of rank Winsorization to that of averaging only observations which lie within small temporal neighborhoods. This leads to a definition of the class of *generalized Wilcoxon* (GW) filters, which are parametrized by three parameters, namely, the degrees of temporal and rank Winsorization and the degree of averaging. The GW filters can be defined to have desirable characteristics of edge preservation, detail retention, and impulse rejection, in addition to the property of Gaussian noise smoothing. Performance characteristics of these filters are considered through analysis and simulations with one-dimensional signals. The filters considered here together with recent results on  $L$ - and  $M$ -filters show that all three well-known classes of robust location estimates, the  $L$ -,  $M$ -, and  $R$ -estimates, can be applied to nonlinear smoothing of signals.

## I. INTRODUCTION

LINEAR filters have been widely used for successfully suppressing additive Gaussian noise in noisy sequences composed of desired signals and noise in many digital signal processing schemes. One of the advantages of linear filters is that efficient FFT algorithms may be exploited to reduce the computation time required for the processing; a wide variety of other efficient and practical hardware implementations are also available for linear filters. It is, however, well known that linear filters, which are best for reducing additive Gaussian noise in a noisy data sequence, give poor performance characteristics in certain situations of practical interest. For example, they smear out any edges in an image, and they are also very poor in suppressing impulsive noise with heavy-tailed probability density function. In these circumstances, nonlinear and/or adaptive techniques need to be employed to

obtain a reasonable recovered version of the desired signal.

In order to overcome these disadvantages of linear filters and to get better performance in such cases, nonlinear techniques have been proposed and shown to be very effective in such situations [1]–[9]. Median filters, whose outputs are defined as medians of the input values in neighborhoods around each point of the discrete-time noisy signal, for example, have strongly nonlinear characteristics, being able to reject quite effectively impulsive noise components while preserving edges in the original signal. These two properties of the median filter have been widely recognized and exploited, and median filters have been rather thoroughly investigated. Examples of applications of median filters can be found in nonlinear speech processing and image processing [10], [11]. Their deterministic as well as statistical properties have also been investigated in [12] and [13]. Efficient realizations for real-time median filtering and VLSI implementations of median filters have been considered [14].

Since the degree of noise smoothing obtainable by a median filter can be influenced only by the processing window size, however, a median filter does not in general allow the user a sufficient degree of control over its characteristics [4]. Furthermore, median filters do not have the averaging operation that is particularly appropriate in reducing additive Gaussian noise components in noisy data; thus, they may perform poorly in Gaussian noise. Therefore, for a better overall performance when the signal has both edges and details and the noise has both Gaussian and impulsive components, it is, in general, highly desirable to implement a filtering scheme with an algorithm that has both nonlinear and linear (averaging) characteristics which can be easily influenced by a set of parameters.

A variety of such techniques, including robust estimation techniques from statistical theory [15]–[18], have quite recently been exploited in the digital signal processing area for this purpose. Cascades of median and linear filters, the order-statistic or  $L$ -filters [3], [4], the  $M$ -filters [4], the  $k$ -nearest neighbor (KNN) filters [1], the symmetric nearest neighbor (SNN) mean filters, the symmetric nearest neighbor (SNN) median filters [2], and the linear median hybrid (LMH) filters [6] are typical examples of such techniques. It has also been shown that nonlinear

Manuscript received December 5, 1987; revised November 30, 1988. This work was supported by the Air Force Office of Scientific Research under Grant 87-0052.

P. P. Gandhi and S. A. Kassam are with the Department of Electrical Engineering, University of Pennsylvania, Philadelphia, PA 19104.

I. Song is with the Department of Electrical and Electronics Engineering, Korea Advanced Institute of Science and Technology (KAIST), P.O. Box 150, Cheongryang, Seoul 130-650, Korea.

IEEE Log Number 8929356.

techniques which inherently employ both linear and nonlinear operations (e.g.,  $M$ -filters and  $L$ -filters) allow the user a useful degree of control over the filter characteristics through a set of variable parameters that may be set *a priori* or adaptively by an appropriate algorithm, allowing better overall performance than median filters [4], [13].

It is noteworthy that, among the filters mentioned above for restoring desired signals from noisy observations, the  $L$ - and  $M$ -filters are based on classes of robust estimates, the  $L$ - and  $M$ -estimates, respectively. Since these two classes of robust estimates from statistical theory have successfully been exploited in the area of digital signal processing, it is quite natural to seek similar applications of the third major class of robust estimates, the rank estimates ( $R$ -estimates), of statistical theory. As an example of the general class of  $R$ -filters, a filter may be based on the Wilcoxon statistic which is commonly used in non-parametric statistics. The Wilcoxon filter selects the median of *averages of pairs* of values (Walsh averages) in each filter window, and thus can be regarded as a generalization of the median filter with an average operation incorporated. Crinon [19] applied the Wilcoxon filter to recover nonstationary signals corrupted with Gaussian noise. It was shown that although the Wilcoxon filter is robust, it smears out the sharp signal transitions. Fast algorithms to compute the Wilcoxon filter output have also been developed in both one-dimensional [19] and two-dimensional [20] filtering applications.

In this paper, a new class of nonlinear discrete-time filters for edge preservation, detail retention, and Gaussian and impulsive noise reduction is considered as an application of a class of rank estimates in signal processing. We introduce what will be called the *generalized Wilcoxon* (GW) filters which are improved versions of the Wilcoxon filter. These filters use both the temporal and rank information of the input signal within a window to produce the output. They will have, in general, better performance characteristics than the mean, median, and Wilcoxon filters. These performance characteristics can be influenced by a set of parameters that can be chosen or varied adaptively, a feature which is common to  $L$ - and  $M$ -filters also. In addition, the design of the GW filters will be independent of the input signal variance.

This paper is organized as follows. In Section II, a general class of  $R$ -filters based on both one- and two-sample rank tests is described along with the basic rationale for their expected usefulness. Some properties of the Wilcoxon and Normal Scores filters and their Winsorized versions are also discussed. The GW filters are then considered in Section III as extensions and improvements of the simple Wilcoxon filter, and their properties are discussed. Two dual classes of nonlinear filters are also defined which include among them most of the interesting nonlinear filters which have been considered in the past. Section IV is devoted to performance analysis and comparisons of the GW filters and other filters through computer simulations.

## II. THE $R$ -FILTERS

One special form of finite impulse response (FIR) linear filtering is obtained when the outputs are running means of the input sequence, computed in processing windows of size  $N$ . As we have noted in the previous section, the median filter may be viewed as its more robust counterpart which is quite effective in impulsive environments but which is not as good for additive Gaussian noise smoothing. The concept of robustness is particularly appropriate in signal processing when the noise has impulsive components, since robustness implies insensitivity to a slight deviation, such as that caused by a small number of impulses, from a nominal assumption (usually of Gaussian noise). Such a robust scheme is also effective for edge preservation since near the start or the end of an edge the data in a window of size  $N$  appears to contain a few impulses. These properties can also be found in the  $L$ - and  $M$ -filters, which are based on the  $L$ - and  $M$ -estimates, respectively, of statistical theory. Let us denote by  $Y$  the output of a filter based on a set of  $N$  observations  $(X_1, X_2, \dots, X_N)$ . The output of an  $L$ -filter is defined to be some linear combination of ordered observations [3], [4]. That is, we have  $Y = \sum_{i=1}^N \beta_i X_{(i)}$  where  $\beta = (\beta_1, \beta_2, \dots, \beta_N)$  is a set of coefficients and  $X_{(i)}$  is the  $i$ th smallest observation in the set  $(X_1, X_2, \dots, X_N)$ . For example, choosing  $\beta = (1, 1, \dots, 1)$  gives us the sample mean filter and  $\beta = (0, \dots, 0, 1, 0, \dots, 0)$  gives us the median filter (for odd  $N$ ). The output of an  $M$ -filter, on the other hand, is defined to be the value of  $Y$  that satisfies  $\sum_{i=1}^N \psi(X_i - Y) = 0$  where  $\psi(\cdot)$  is an odd function [4]. The  $M$ -filter reduces to the sample mean filter if  $\psi(x) = x$  and to the median filter as  $\psi(x) \rightarrow \text{sgn}(x)$ .

The essence of the operation of an  $R$ -filter is that more importance is placed on *ranks* of the data in forming the output for each window; the output of the  $R$ -filter is determined using the relative ranks of the data in each processing window at any time index. Thus, extreme values such as those produced by impulsive components can be made to have much less of an effect on the output than would be the case in linear filters. For example, the sequence  $\{2.1, 3.2, 4.3, 5.9, 0.9\}$  will appear to be the same as the sequence  $\{2.1, 3.2, 4.3, 59.0, 0.9\}$  to a properly designed  $R$ -filter, because both sequences have the rank vector  $\{2, 3, 4, 5, 1\}$ . Furthermore, when the processing window is located near the start or the end of an edge, the edge appears to produce a few impulsive components, and therefore the  $R$ -filter, in general, does not smear an edge as much as a linear filter. These observations of insensitivity of  $R$ -filters to impulses can also be made for  $L$ - and  $M$ -filters. In fact, the  $L$ -filter also explicitly uses rank information in weighting data values.

Let us consider a vector  $X$  of  $N$  observations  $(X_1, X_2, \dots, X_N)$  governed by cumulative distribution function (cdf)  $F(x - \theta)$  and assume that the probability density is symmetric about the location parameter  $\theta$ . We will denote the ordered observations by  $X_{(1)}, X_{(2)}, \dots, X_{(N)}$  with

$X_{(1)} \leq X_{(2)} \leq \dots \leq X_{(N)}$ . In general, the rank estimate of  $\theta$  is defined from rank tests for  $\theta = 0$  versus the alternative  $\theta > 0$ . In this section, we will present rank estimators based on both the one- and two-sample rank tests for location. We will, however, restrict our attention to filters based only on the one-sample rank tests.

*A. One-Sample Rank Filters*

A general *one-sample rank test* of location for testing  $H: \theta = 0$  versus  $\bar{H}: \theta > 0$  is given by<sup>1</sup>

$$\phi(X) = \begin{cases} 1 & T > c \\ 0 & T \leq c \end{cases} \quad (2.1)$$

where  $\phi(X)$  is a test function and  $T$  is a test statistic. Here the test statistic  $T$  depends on the ranks of the observations. The constant  $c$  determines the rejection region of  $H$  and is found for some prescribed type-I error  $\alpha$ , which is

$$E_{\theta=0}[\phi(X)] = P[T > c | H] = \alpha \quad (2.2)$$

where  $0 \leq \alpha < 1$ . In particular, the test statistic  $T$  is given by [17]

$$T = \sum_{i=1}^N a \left[ \frac{R(|X_i|)}{N+1} \right] u(X_i) \quad (2.3)$$

where  $R(|X_i|) = R_i$  is the rank of the *ith absolute value*  $|X_i|$  in the set of absolute values  $\{|X_1|, |X_2|, \dots, |X_N|\}$ , the *score function*  $a(\cdot)$  is a nondecreasing weighting function defined on  $(0, 1)$ , and  $u(\cdot)$  is the unit step function. This statistic is called the one-sample *signed rank* statistic. In order to compute the value of  $T$  for a given set of observations, we need only  $N$  uniformly spaced score values of the score function. These  $N$  scores will be denoted by  $a_i$  where  $a_i = a\{i/(N+1)\}$ ,  $i = 1, 2, \dots, N$ , and  $a_0 \equiv 0$ . The test statistic  $T$  given by (2.3) is a reasonable statistic for testing  $H$  versus  $\bar{H}$ , because under  $H$  any of the ranks  $R_i$  can correspond to either positive or negative observations with equal probability. It is easy to see that the distribution of  $T$  under  $H$  is symmetric about  $\sum_{i=1}^N a_i/2$ . Under the alternative  $\bar{H}$ , however, the cdf of any observed sample is translated to the right by amount  $\theta$ . Hence, larger ranks that correspond to positive observations occur with higher probability than those corresponding to negative observations. This leads to a larger value for  $T$  and the null hypothesis is rejected.

For estimation purposes, the observations are assumed to be governed by a symmetric probability density function with location parameter  $\theta$ , which is the parameter to be estimated. To estimate  $\theta$ , we modify (2.3) to define a

<sup>1</sup>The alternative hypothesis  $\bar{H}$  may also be taken to be  $\theta \neq 0$  or  $\theta < 0$ . For instance, for  $\bar{H}: \theta \neq 0$ , the modified rank test becomes  $\phi(X) = 1$  if  $|\hat{T}| > c$  and  $= 0$  if  $|\hat{T}| \leq c$  where  $\hat{T} = \sum_{i=1}^N a[R(|X_i|)/N+1] \text{sgn}(X_i)$ . Here the distribution of  $\hat{T}$  is symmetric about 0. We are interested in the problem of signal restoration or *estimation*, and the one-sample rank test for any of the three alternatives will lead to the same result for signal estimation.

quantity  $T(\theta)$  given by

$$T(\theta) = \sum_{i=1}^N a \left[ \frac{R(|X_i - \theta|)}{N+1} \right] u(X_i - \theta), \quad (2.4)$$

where  $R(|X_i - \theta|) = R_i(\theta)$  is the rank of the *ith absolute shifted value*  $|X_i - \theta|$  in the set  $\{|X_1 - \theta|, |X_2 - \theta|, \dots, |X_N - \theta|\}$ . Note that the distribution of  $T(\theta)$  is symmetric about  $\sum_{i=1}^N a_i/2$ . We now define the estimate  $\hat{\theta}$  of the unknown location  $\theta$  as that value for which the statistic  $T(\hat{\theta})$  takes on the center value of the distribution of  $T(\theta)$ . In other words,  $\hat{\theta}$  is the solution of the implicit equation

$$T(\hat{\theta}) = \sum_{i=1}^N \frac{a_i}{2}. \quad (2.5)$$

The equality may not be always achievable due to the discrete nature of  $T(\hat{\theta})$ . Hence,  $\hat{\theta}$  is in practice taken to be the value of  $\theta$  for which  $T(\hat{\theta})$  is closest to  $\sum_{i=1}^N a_i/2$ . The location estimate defined by (2.5) is known as the *Hodges-Lehmann estimate* [21] of the location  $\theta$ .

A useful alternative expression of  $T(\theta)$  is available [22] in terms of the difference of *adjacent scores*. This is

$$T(\theta) = \sum_{i=1}^N \sum_{\substack{j=1 \\ i \leq j}}^N (a_{j-i+1} - a_{j-i}) u(W_{(ij)} - \theta). \quad (2.6)$$

Here the *ordered Walsh average*  $W_{(ij)}$  is defined as the average  $(X_{(i)} + X_{(j)})/2$  of the ordered observations  $X_{(i)}$  and  $X_{(j)}$ . We will also use the notation  $W_j$  to mean the pairwise Walsh average  $(X_i + X_j)/2$ . As a special case, consider a set of scores  $\{a_1, a_2, \dots, a_N\}$  for which the steps  $a_{j-i+1} - a_{j-i} = C$  or zero, where  $C$  is a positive constant. Then the estimate  $\hat{\theta}$  based on the signed rank quantity  $T(\theta)$  is given by

$$\hat{\theta} = \text{median} \{W_{(ij)} : (i, j) \in S\} \quad (2.7)$$

where the set  $S$  is defined by

$$S = \{(i, j) : a_{j-i+1} - a_{j-i} = C, \quad 1 \leq i \leq j \leq N\}. \quad (2.8)$$

The condition that the *difference* of the two adjacent scores must either be constant or zero implies that the score function is either piecewise linearly increasing with slope  $C$  or piecewise constant. The Hodges-Lehmann estimate, in this case, assumes a very simple form given in (2.7). Note also that (2.7) represents a generalization of median filters with an inherent averaging operation, a property which has been one of our objectives to obtain as a desirable one.

Before we proceed further in discussing the  $R$ -estimates, let us briefly consider the relationships between the three common classes of robust estimates. In order to see the similarities between  $L$ - and  $R$ -estimates, let us substitute for the unit step function  $u(\cdot)$  an equivalent

signum function in (2.6) to get

$$T(\theta) = \sum_{i=1}^N \sum_{\substack{j=1 \\ i \leq j}}^N (a_{j-i+1} - a_{j-i}) \cdot \left\{ \text{sgn}(W_{(ij)} - \theta) + 1 \right\} / 2. \quad (2.9)$$

The location estimate  $\hat{\theta}$  is the solution of

$$\begin{aligned} & \sum_{i=1}^N \sum_{\substack{j=1 \\ i \leq j}}^N (a_{j-i+1} - a_{j-i}) \left\{ \text{sgn}(W_{(ij)} - \hat{\theta}) + 1 \right\} / 2 \\ & = \sum_{i=1}^N a_i / 2. \end{aligned} \quad (2.10)$$

Simplifying (2.10) we get

$$\sum_{i=1}^N \sum_{\substack{j=1 \\ i \leq j}}^N (a_{j-i+1} - a_{j-i}) \text{sgn}(W_{(ij)} - \hat{\theta}) = 0. \quad (2.11)$$

Now if we replace the signum function with a linear function, it is easy to see that the location estimate  $\hat{\theta} = \hat{\theta}_L$  is given by

$$\hat{\theta}_L = \sum_{i=1}^N \beta_i X_i, \quad (2.12)$$

where

$$\beta_i = \frac{a_i + a_{N-i+1}}{2 \sum_{i=1}^N a_i}. \quad (2.13)$$

Note that  $\hat{\theta}_L$  satisfies an unbiasedness condition since  $\beta_i = \beta_{N-i+1}$ , and  $\sum_{i=1}^N \beta_i = 1$ . The  $R$ -estimate resulting from (2.4) or (2.5) may be considered as a restricted  $L$ -estimate in the sense that only the sign of  $W_{(ij)} - \hat{\theta}$  is used in determining the  $R$ -estimate. A similar relationship also exists between the  $R$ - and  $M$ -estimates. For example, if we consider a linear score function in (2.4) and use the result (2.5), we get [19]

$$\sum_{i=1}^N R(|X_i - \hat{\theta}|) \text{sgn}(X_i - \hat{\theta}) = 0. \quad (2.14)$$

Now an  $M$ -estimate is defined as a solution of  $\sum_{i=1}^N \psi(X_i - \hat{\theta}_M) = 0$ , for an appropriately defined  $\psi$  function. Thus,  $\hat{\theta}$  in (2.14) may be interpreted as a modified  $M$ -estimate. For an arbitrary score function, the  $R$ -estimate resulting from (2.4) and (2.5) can similarly be considered as a modified  $M$ -estimate.

In order to define an  $R$ -filter, let us consider a time series  $\{Z_i\}$  which is to be filtered. The filters we consider here will act as finite moving window filters of window size  $2n + 1$ , and will process samples  $\{Z_{k-n}, Z_{k-n+1}, \dots, Z_{k+n}\}$  to produce the output  $Y_k$  at the  $k$ th time instant. For convenience we will denote this finite window sequence by  $(X_1, X_2, \dots, X_N)$  where  $X_i = Z_{k-n+i-1}$  for

$i = 1, 2, \dots, N$  and  $N = 2n + 1$ . Then the output of the  $R$ -filter at the  $k$ th time instant is given by the Hodges-Lehmann estimate  $\hat{\theta}$  based on  $(X_1, X_2, \dots, X_N)$ . Note that the  $R$ -filter is based only on the rank ordered set  $\{X_{(1)}, X_{(2)}, \dots, X_{(N)}\}$  and ignores temporal ordering, as do the  $L$ - and  $M$ -filters. The output  $Y_k$  of the  $R$ -filter is simply the  $R$ -estimate of the location of the input sequence in the window.

Even though a general formulation of  $R$ -filters can be expressed by (2.4) and (2.5), it is difficult to get more insights and to deduce detailed characteristics from the above formulation because  $a(\cdot)$  has not yet been given explicitly. The score function  $a(\cdot)$  plays an important role in the  $R$ -filter, determining its performance characteristics. By choosing appropriate forms for  $a(\cdot)$ , a number of special  $R$ -filters can be defined with different specific characteristics. Let us now look at some examples of  $R$ -filters that may be derived from the general framework presented above.

*Example 1 (Median Filter):* Suppose the score function is constant on  $(0, 1)$  as shown in Fig. 1(a). The test resulting from such a score function is known as the *sign test*. For this case we have  $a_{j-i+1} - a_{j-i} = 1$  for  $j = i$  and 0 for  $j > i$ . Recall that  $a_0 = 0$ . The estimate  $\hat{\theta}$  is simply given by the *median* of  $N$  observations, and the filter output is

$$Y_k = \text{median} \{X_{(1)}, X_{(2)}, \dots, X_{(N)}\}. \quad (2.15)$$

For odd  $N$ , median is the center value of the ordered observations, and for even  $N$ , it is the average of the two center values of the ordered observations. It is well known that the median filter preserves edges and removes impulses. ♦

The following examples of  $R$ -filters exhibit tradeoffs between Gaussian noise smoothing and edge performance in the class of general  $R$ -filters.

*Example 2 (Wilcoxon Filter):* Consider a linearly increasing score function as shown in Fig. 1(b). Here the difference in the adjacent scores is the slope of the score function. The filter output, in this case, is given by

$$Y_k = \text{median} \{W_{(ij)} : 1 \leq i \leq j \leq N\} \quad (2.16a)$$

$$= \text{median} \{W_{ij} : 1 \leq i \leq j \leq N\}. \quad (2.16b)$$

The Wilcoxon filter consists of both linear and nonlinear operations; however, its ability to preserve edges is not very good [19]. This is because the Walsh averages are computed over all pairs of indexes  $(i, j)$  such that  $i \leq j$ . It is easy to see that the Wilcoxon filter computes the median of  $N(N + 1)/2$  Walsh averages, which is a significant increase in number compared to the median filter. For large values of  $N$ , the Wilcoxon filtering output computation can be quite demanding. Efficient algorithms have been explored to reduce this computational burden in both one- and two-dimensional Wilcoxon filters [19], [20].

Let us now consider the edge response of the Wilcoxon filter. Suppose the input to the Wilcoxon filter is an ideal

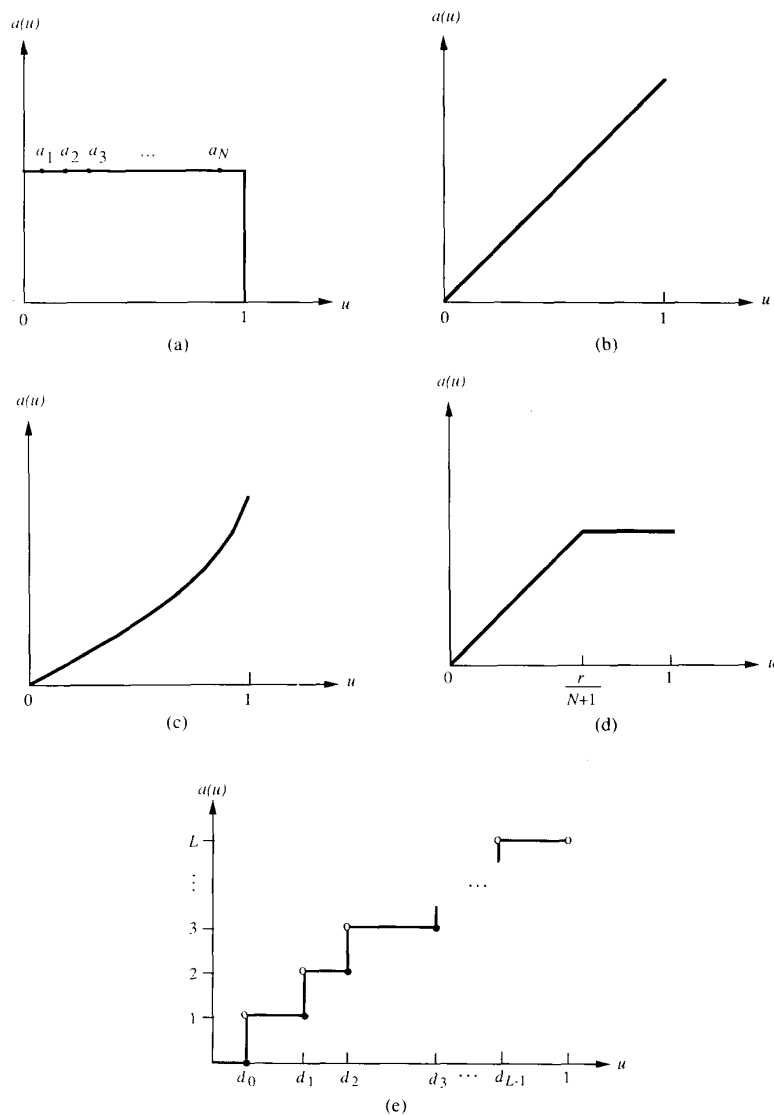


Fig. 1. One-sample score functions. (a) Median. (b) Wilcoxon. (c) Normal scores. (d) Winsorized Wilcoxon. (e) Modified median.

edge. An ideal edge is noise free and may be defined as a step from a constant value to another constant value. We will assume that the step is from 0 to 1. Let us consider an input with an ideal edge which can be represented by the sequence  $\{\dots 0, 0, 0, 0, 1, 1, 1, 1, \dots\}$ . It is quite straightforward to obtain the filter output. For example, for  $N = 5$  and 7, the output of the Wilcoxon filter is  $\{\dots 0, 0, 0, 1/2, 1/2, 1, 1, 1, \dots\}$ , and for  $N = 9$  and 13 it is  $\{\dots 0, 0, 1/2, 1/2, 1/2, 1/2, 1, 1, \dots\}$ . This shows that the Wilcoxon filter will generally smear edges and distort narrow pulses, but not quite so badly as a running mean filter would.

Although we are not really interested in the asymptotic performance (for  $N \rightarrow \infty$ ) of our filters, a consideration of asymptotic performance can allow some interesting comparisons to be made. The asymptotic performance of

one filter relative to another filter can be measured in terms of the *asymptotic relative efficiency* (ARE). There are many different ways of defining the ARE. Consider two estimates  $\hat{\theta}_1$  and  $\hat{\theta}_2$  of the location  $\theta$  based on  $N$  observations, with the asymptotic distribution of  $N^{1/2}[\hat{\theta}_i - \theta]$  approaching the zero-mean normal distribution with variance  $\tau_i^2$ . We define the ARE of  $\hat{\theta}_2$  with respect to  $\hat{\theta}_1$  as [23]

$$e_{2,1} = \tau_1^2 / \tau_2^2. \tag{2.17}$$

The median, for example, is approximately 67 percent efficient compared to the sample mean if the observations come from a Gaussian distribution. The asymptotic performance of the Wilcoxon filter is very similar to that of the mean filter. In fact, the Wilcoxon filter is approximately 96 percent efficient compared to the mean filter

under the assumption of Gaussian noise. Hence, it is advantageous to use the Wilcoxon filter for filtering slowly varying signals corrupted by additive Gaussian and impulsive noise. The Wilcoxon filter performance in such cases is generally superior to that of the mean filter since it removes the impulsive noise components effectively without compromising the amount of Gaussian noise smoothing (as would be the case in the median filter). The Wilcoxon filter, whose output is expressed by (2.16b), has been investigated in [19]. The study in [19] appears to be the first published study on applications of  $R$ -estimates to signal restoration. ♦

*Example 3 (Normal Scores Filter):* The Normal Scores (NS) filter is obtained by letting  $a_i = \Phi^{-1}\{1/2 + i/2(N + 1)\}$ ,  $i = 1, 2, \dots, N$  with  $\Phi^{-1}(\cdot)$  being the inverse of the standard normal distribution as depicted in Fig. 1(c). This score function does not satisfy the condition that differences of adjacent scores are either constant or zero. Therefore, the resulting filter output does not lead to a closed-form expression as in (2.7). One may use numerical techniques to implement the NS filter and any other filter with a general score function based on (2.5). The NS filter is asymptotically as efficient as the sample mean and slightly more efficient than the Wilcoxon filter for additive white Gaussian noise. However, the edge performance of the NS filter is inferior to that of the Wilcoxon filter. For example, the output of the NS filter due to an ideal edge is  $\{\dots 0, 0, 1/2, 1/2, 1/2, 1/2, 1, 1, \dots\}$  if  $N = 7$  is used. Since the output of the NS filter is determined by locating the point of symmetry of (2.4), the computation burden becomes rather intensive. This burden can be reduced if a smaller window size is used. However, our simulation results indicate that the NS filter performance for relatively small windows is not much different from that of the Wilcoxon filter. ♦

*Example 4 (Winsorized Rank Filter):* As discussed above, the edge performance of a rank filter will generally be poor if the difference of adjacent scores is different from zero for larger values of the score index. In this case, the set of ordered Walsh averages contains averages of values across the edge. It may be possible to improve the edge performance of such an  $R$ -filter if we limit the distance over which the Walsh averages are performed. This is equivalent to limiting the score function beyond some index  $r$  to a constant value. That is, data values with ranks larger than some maximum value are treated with uniform weights. The process of limiting of the score function after some distance  $r$  is known as *Winsorization*. The median filter is an extreme example of such an  $R$ -filter where the entire score function is Winsorized to give an excellent edge response.

Winsorization can be applied, for example, to the NS filter, to improve the edge performance and to reduce the computation burden. A more interesting filter can be defined if the Wilcoxon filter is Winsorized. The score function in this case incorporates a mixture of both the Wilcoxon-type (linear) and the median-type (constant) score functions. This is achieved by using a score function that

is similar to the Wilcoxon score function on  $(0, r)$  and constant on  $[r, 1)$  as shown in Fig. 1(d). The resulting statistic  $T$  is called the *Winsorized Wilcoxon signed rank statistic* with  $a_i = \min(i, r)$ . The corresponding output of the *Winsorized Wilcoxon filter* is given by

$$Y_k = \text{median} \{W_{(ij)}; j - i < r, \quad 1 \leq i \leq j \leq N\}. \quad (2.18)$$

The edge performance of this filter is enhanced since the set of ordered Walsh averages is restricted to contain  $W_{(ij)}$  satisfying  $j - i < r$  for some maximum distance  $r$ . By letting  $r$  range from 1 to  $N$ , the Winsorized Wilcoxon filter ranges from the median filter to the Wilcoxon filter. The improvement in edge performance is seen by considering an ideal edge at the input. The corresponding output of the Winsorized Wilcoxon filter for  $N = 9$  is  $\{\dots 0, 0, 0, 1/4, 3/4, 1, 1, 1, \dots\}$  if  $r = 3$  and is  $\{\dots 0, 0, 0, 1/2, 1/2, 1, 1, 1, \dots\}$  if  $r = 4, 5$ , and 6. The smearing of the edge is clearly reduced compared to the Wilcoxon filter with the same window size. Hence, the Winsorized Wilcoxon filter offers a compromise in performance between the Wilcoxon and the median filters. The value of the parameter  $r$  can be chosen between the two extremes to obtain better edge performance compared to the Wilcoxon filter and better smoothing compared to the median filter.

Clearly, the computation burden is reduced in the Winsorized Wilcoxon filter compared to the Wilcoxon filter if  $r$  is chosen to be less than  $N$ . In general, the total number of ordered Walsh averages over which the median operation is performed is given by

$$\begin{aligned} M_e &= (N - r + 1)r + \sum_{i=N-r+2}^N (N - i + 1), \\ &1 \leq r \leq N \\ &= (2N - r + 1)r/2. \end{aligned} \quad (2.19)$$

Note that  $M_e$  is reduced to  $N$  if  $r = 1$  (median filter) and to  $N(N + 1)/2$  if  $r = N$  (Wilcoxon filter). ♦

*Example 5 (Modified Median Filter):* In this example, we describe an  $R$ -filter based on an  $L$ -step score function [24]. This score function is shown in Fig. 1(e) and is described by

$$a(u) = \begin{cases} 0 & 0 < u \leq d_0/(N + 1), \\ 1 & d_0/(N + 1) < u \leq d_1/(N + 1), \\ \vdots & \vdots \\ L - 1 & d_{L-2}/(N + 1) < u \leq d_{L-1}/ \\ & (N + 1), \\ L & d_{L-1}/(N + 1) < u < 1 \end{cases} \quad (2.20)$$

where  $0 \leq d_0 \leq d_1 \leq \dots \leq d_{L-1} \leq N$  are integers. Let us denote by  $S(m)$  the set of indexes over which the

ordered Walsh averages are computed for the  $m$ th discontinuity in the score function. In other words,

$$S(m) = \{(i, j) : j - i = d_m, \quad 1 \leq i \leq j \leq N\}. \quad (2.21)$$

Hence,

$$\hat{\theta} = \text{median} \left\{ W_{(ij)} : (i, j) \in \bigcup_{m=0}^{L-1} S(m) \right\}. \quad (2.22)$$

Note that the modified median filter reduces to the Winsorized Wilcoxon filter if  $L = r$  and  $d_m = m$  for  $m = 0, \dots, r-1$ ; for  $r = 1$  we get the median filter.  $\blacklozenge$

Both the Wilcoxon and the NS filters are robust compared to the sample mean in the sense that their performance is not affected by a few impulses present within a particular window. However, both filters smear out edges. The Winsorized Wilcoxon and the Winsorized NS filters do offer some improvement in the edge performances. In Section IV we will discuss some simulation results illustrating this.

### B. Two-Sample Rank Filters

In this subsection we discuss the rank estimate resulting from a two-sample rank test. Although we will only consider filters based on the one-sample rank estimates in the rest of this paper, we have, nevertheless, provided a rather detailed treatment of rank filters based on the two-sample location estimate for the sake of completeness. Two-sample rank tests have early on been applied to construct one-sample estimators in nonparametric statistics. The one-sample rank estimates we have described turn out to be closely related to those constructed from the two-sample rank test starting point. The second sample in this application is artificially generated as explained below by separating data values with different signs. Consider first two random samples,  $(U_1, U_2, \dots, U_N)$  from cdf  $F(x - \theta_u)$  and  $(X_1, X_2, \dots, X_N)$  from cdf  $F(x - \theta_x)$ . Let us define  $\theta = \theta_x - \theta_u$  to be the location difference of the two samples. A general two-sample statistic for testing location shift between the two samples may be defined as [18]

$$W = \sum_{i=1}^N b \left[ \frac{R_i}{2N+1} \right] \quad (2.23)$$

where  $b(\cdot)$  is a nondecreasing score function defined on  $(0, 1)$  with  $b(u) = -b(1-u)$ , and  $R_i$  is the rank of the  $i$ th observation from the second sample of size  $N$  (the rank of  $X_i$ ) in the pooled sample of size  $2N$ . Note that  $b(u)$  is odd-symmetric about the point  $u = 1/2$ .

If the locations of the two samples are the same, the ranks of the observations from the second sample (a sample of size  $N$ ) are equally likely to be any set of  $N$  of the first  $2N$  integers. On the other hand, if the location of the second sample is beyond that of the first sample, the ranks of the observations from the second sample will tend to have larger values, resulting in a larger value for  $W$  on the average. Therefore, by comparing  $W$  to a threshold, a

test for location difference for the two samples is obtained.

To estimate the difference in location  $\theta$ , (2.23) is modified to become

$$W(\theta) = \sum_{i=1}^N b \left[ \frac{R_i(\theta)}{2N+1} \right] \quad (2.24)$$

where  $R_i(\theta)$  is the rank of  $X_i - \theta$  among the pooled sample  $(U_1, \dots, U_N, X_1 - \theta, \dots, X_N - \theta)$ . Following a procedure similar to that in the one-sample case,  $W(\theta)$  of (2.24) can alternatively be expressed in terms of the differences  $\Delta_{(ij)}$  as [17]

$$W(\theta) = Q(\theta) + \sum_{i=1}^N b_i \quad (2.25)$$

where

$$Q(\theta) = \sum_{i=1}^N \sum_{j=1}^N (b_{j+i} - b_{j+i-1}) u(\Delta_{(ij)} - \theta),$$

$$b_i = b \left( \frac{i}{2N+1} \right), \quad (2.26)$$

and

$$\Delta_{(ij)} = X_{(i)} - U_{(j)}. \quad (2.27)$$

As in the one-sample case, we may similarly describe the estimate  $\hat{\theta}$  of the location difference  $\theta$  for the special case of a score function that is either piecewise constant or linearly increasing in the following way:

$$\hat{\theta} = \text{median} \{ \Delta_{(ij)}, \quad (i, j) \in S \} \quad (2.28)$$

where

$$S = \{(i, j) : b_{j+i} - b_{j+i-1} = \text{constant}, \\ 1 \leq i, j \leq N\}.$$

For an arbitrary score function, it is generally not possible to compute  $\hat{\theta}$  based on (2.28). In this case,  $\hat{\theta}$  can be computed from the implicit equation [18]

$$W(\hat{\theta}) \approx 0 \quad (2.29)$$

where  $W(\theta)$  is given by (2.24). In essence, (2.29) implies that the value  $\hat{\theta}$  should be found so that the resulting two sets of ranks of the two samples defined above, from the pooled sample, are statistically similar.

As an interesting observation, we note that one direct application of the two-sample rank test is in edge detection. Consider a data set of length  $2N$  with a possible edge in the middle. Let us denote the leftmost  $N$  data values by  $(U_1, U_2, \dots, U_N)$  and the rightmost  $N$  values by  $(X_1, X_2, \dots, X_N)$ . We may now compute the value of the test statistic  $W$  of (2.23) for a given score function and compare it to some threshold  $\tau$ . The value of  $\tau$  can be set for a desired value of the probability of false alarm (type I error), and an edge is detected if  $W$  exceeds the threshold.

Let  $Y_k$  be the candidate location estimate for the sample of size  $N$  centered on the  $k$ th time index. Since only one sample  $\{X_i\}$  of size  $N$  is available in each window in our problem of signal restoration, another set  $\{U_i\}$  has to be defined from each of the original samples of size  $N$  before further processing in which such a rank statistic is used. For this purpose, an image of the sample in each window at a candidate location value  $Y_k$  may be employed. To do this, we center the original sample  $(X_1, \dots, X_N)$  to become  $(X_1 - Y_k, \dots, X_N - Y_k)$ , and then invert signs to get  $(Y_k - X_1, \dots, Y_k - X_N)$  as the second sample. The  $X$  and the desired  $U$  samples may now be inserted in (2.24) and (2.29) [or (2.28)] to obtain an estimate of their location difference, which would then be forced to be zero or close to zero. Substituting  $Y_k - X_j$  for  $U_j$  in (2.28) with  $Y_k = \hat{\theta}$ , we get

$$Y_k = \text{median} \{W_{(i, N-j+1)} : (i, j) \in S\}. \quad (2.30)$$

A two-sample score function  $b(u)$  can be generated from its one-sample counterpart  $a(u)$  by setting  $b(u) = -a(1 - 2u)$  if  $0 < u \leq 1/2$  and  $a(2u - 1)$  if  $1/2 < u < 1$ . The two-sample score functions generated from the one-sample counterparts for the median and Wilcoxon filters are shown in Fig. 2(a) and (b), respectively. Let us now consider some examples of  $R$ -filters resulting from the two-sample rank tests.

*Example 6 (Two-Sample Median Filter):* The score function for a two-sample median filter is shown in Fig. 2(a). It is evident from the figure that  $b_{j+i} - b_{j+i-1} = 2$  if  $j + i = N + 1$ . The output of the filter is given by

$$Y_k = \text{median} \{X_{(1)}, X_{(2)}, \dots, X_{(N)}\}. \quad (2.31)$$

This follows directly from (2.28) and is the same as in the one-sample case. ♦

*Example 7 (Two-Sample Wilcoxon Filter):* As a second example, consider a two-sample Wilcoxon filter for which the score function is linear and is given by  $b(u) = 2u - 1$  as shown in Fig. 2(b). In this case, it can be shown that (2.28) can be explicitly solved to give

$$Y_k = \text{median} \{W_{ij} : 1 \leq i, j \leq N\}. \quad (2.32)$$

This should be compared to (2.16b). Here the median operation is performed over *all*  $N^2$  Walsh averages compared to only  $N(N + 1)/2$  Walsh averages required in (2.16b). Even with this additional computational burden, the output of the two-sample Wilcoxon filter in many cases is quite similar to that of the one-sample Wilcoxon filter. In fact, it can be shown that the two-sample Wilcoxon filter is asymptotically equivalent to the one-sample Wilcoxon filter [18]. ♦

More generally, filters or estimates based on one-sample and corresponding two-sample rank tests will be slightly different in their characteristics. However, these differences are not very marked, and we will focus only on filters based on the one-sample procedures in the rest of this paper.

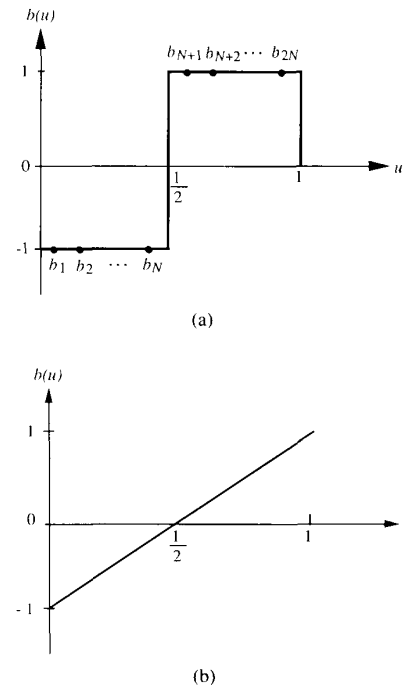


Fig. 2. Two-sample score functions. (a) Median. (b) Wilcoxon.

### III. GENERALIZED RANK FILTERS

We have seen in the previous section that Winsorization of the score functions plays an important role in terms of preserving edges or fine details. In particular, the Winsorized versions of either the Wilcoxon or the NS filters may be used to filter nonstationary input sequences containing edges and outliers with better smoothing of the background Gaussian noise as compared to the median filter. Such Winsorization is achieved simply by limiting the score function to a constant value for larger ranks.

The  $R$ -filters that we have described so far use only the rank-ordered observations to produce the output sequence. For example, the  $R$ -filter does not discriminate between a pulse of width  $w$  and two narrow pulses of width  $w/2$  if they lie within a single window, since the ordered sequence for both cases is the same. For example, consider the original signal shown in Fig. 3(a) which contains two edges enclosing a pulse of width 5 on the left and a pulse of width 4 on the right. The corresponding noisy signal is shown in Fig. 3(b) with additive zero-mean Gaussian noise of unity variance and a deliberately placed impulse between the narrow feature of width 4 and the right edge. Using different filtering schemes with  $N = 9$ , we see from Fig. 3(c)–(g) that we are unable to recover the signal with these two features. For such situations, it would be useful to have a class of filters that incorporate not only the ranks but also the temporal positions of the input data sequence in producing the outputs. In the example just considered, this would allow use of the data values in pairs or groups which are close neighbors based on temporal-order and rank-order information. This leads to a concept of Winsorization in both temporal and rank



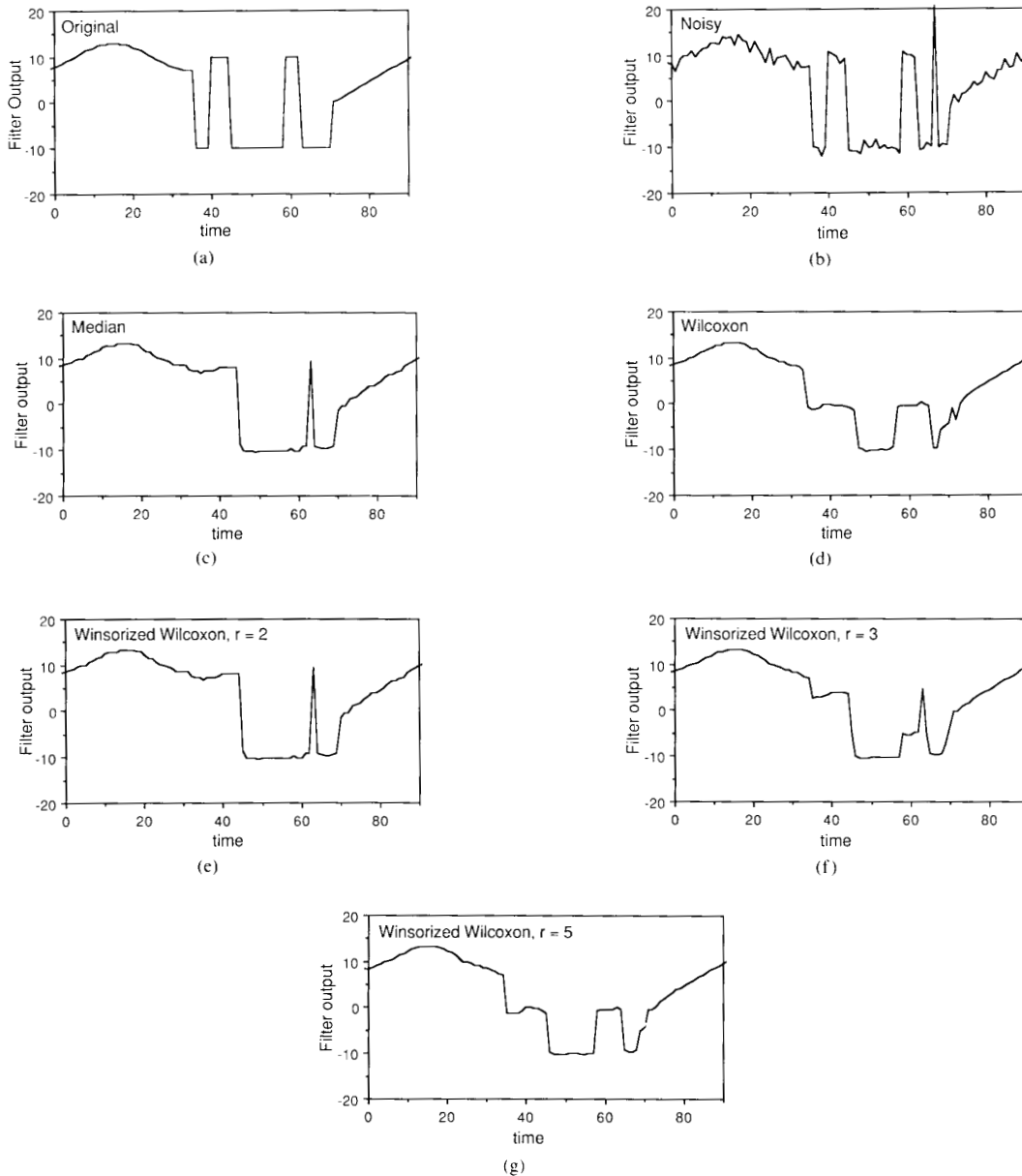


Fig. 3. Simulation results with  $N = 9$ . (a) Original signal. (b) Noisy signal. (c) Output of the median filter. (d) Output of the Wilcoxon (or NS) filter. (e) Output of the Winsorized Wilcoxon filter with  $r = 2$ . (f) Output of the Winsorized Wilcoxon filter with  $r = 3$ . (g) Output of the Winsorized Wilcoxon filter with  $r = 5$ .

domains; that is, the output is computed based on adjacency in time and rank positions of the windowed sequence. Such filters may be described as doubly Winsorized filters. We will use the concepts of  $R$ -estimates of Section II to define filters based on time-rank adjacency.

*A. Filters Based on Temporal Score Functions*

Before we describe a more general class of filters based on time-rank Winsorization, let us first consider a direct counterpart of the rank-order filters in the temporal do-

main. Specifically, consider the Wilcoxon filter given by (2.16a) and (2.16b). Note that the filter output in this case is not affected by whether or not the input data values are sorted. The Winsorized Wilcoxon filter given by (2.18) is based on (2.16a). Here, the median is computed over pairwise averages of ordered values that are in proximity in terms of *ranks*. Similarly, we may define a different filter based on (2.16b) which computes the median over pairwise averages that are not too far apart *temporally*. We will refer to this class of filters as time-Winsorized

filters (as opposed to the rank-Winsorized filters of Section II).

In order to formally define a class of time-Winsorized filters corresponding to arbitrary score functions, let us consider the quantity  $P(\theta)$  defined as

$$P(\theta) = \sum_{i=1}^N \sum_{\substack{j=1 \\ i \leq j}}^N (\bar{a}_{j-i+1} - \bar{a}_{j-i}) u(W_{ij} - \theta) \quad (3.1)$$

where  $\bar{a}(\cdot)$  is the temporal score function defined on  $(0, 1)$  with scores  $\bar{a}_i = \bar{a}[i/(N+1)]$ . Then it is shown in Appendix A that  $P(\theta)$  is symmetrically distributed about  $\sum_{i=1}^N \bar{a}_i/2$ .

To compute the estimate  $\hat{\theta}$  based on (3.1), set  $P(\hat{\theta}) = E[P(\theta)]$  as before. In order to get a closed-form expression for  $Y_k = \hat{\theta}$ , we restrict the temporal score function to be either piecewise linearly increasing with slope  $\tilde{C}$  or piecewise constant. Then we get, as a counterpart of the result (2.7),

$$Y_k = \text{median} \{W_{ij} : (i, j) \in \tilde{S}\} \quad (3.2)$$

where the set  $\tilde{S}$  of indexes  $(i, j)$  is given by

$$\tilde{S} = \{(i, j) : \bar{a}_{j-i+1} - \bar{a}_{j-i} = \tilde{C}, \quad 1 \leq i \leq j \leq N\}. \quad (3.3)$$

The filter defined by (3.2) is simply the time domain version of (2.7). It is easily seen that both (3.2) and (2.7) yield the same output value if the input data sequence is monotonic. Hence, both filters will have the same ideal edge response. Furthermore, (3.2) agrees with (2.7) if the temporal score function is either constant [median-type, see (2.15)] or linearly increasing [Wilcoxon-type, see (2.16b)].

A more interesting filter results when the temporal score function is constrained to be linearly Winsorized [see Fig. 1(d)]. The filter output in this case may be written as

$$Y_k = \{W_{ij} : j - i < t, \quad 1 \leq i \leq j \leq N\} \quad (3.4)$$

where  $t$  is the degree of temporal Winsorization. We call this filter the *time-Winsorized Wilcoxon* filter for obvious reasons. The rank- and time-Winsorized Wilcoxon filters generally have similar performances. Both filters reduce to either a median or a Wilcoxon filter depending on the degree of Winsorization, and they both have identical ideal edge response. The time-Winsorized Wilcoxon filter, however, does enjoy some advantage over its counterpart in the presence of two closely separated narrow pulses as we pointed out at the beginning of this section. We shall elaborate on this particular issue in detail in the following subsections and in Section IV. In addition, although the number of pairwise averages over which the median is computed is the same for both filters, the rank-Winsorized Wilcoxon filter requires additional computation of the order of  $N \log_2(N)$  since it uses the ordered values of the input data. Hence, for larger window sizes, the time-Winsorized filters are computationally less intensive compared to the rank-Winsorized filters.

## B. The Generalized Wilcoxon Filter

In this subsection we introduce a filtering technique that combines Winsorization in the temporal and rank domains. As we pointed out before, the Walsh averages play an important role in  $R$ -filters. Hence, it is natural to define a *Generalized Wilcoxon* (GW) filter that employs Walsh averages computed only for data points which are within fixed temporal and rank distances of each other. The output  $Y$  of the GW filter is defined as

$$Y = \text{median} \{W_{ij} : |R_j - R_i| < r \text{ and } j - i < t, \\ 1 \leq i \leq j \leq N\} \quad (3.5)$$

where  $R_i$  is the rank of  $X_i$ , and  $1 \leq t, r \leq N$  are integers. We will refer to  $t$  and  $r$  as the degree of time-Winsorization and the degree of rank-Winsorization, respectively. It is clear from the definition (3.5) that the set of  $W_{ij}$  over which the median operation is performed is nonempty for any allowable values of  $t$  and  $r$  (between 1 and  $N$ ), since this set always contains the self-terms  $X_i (= W_{ii})$ ,  $i = 1, 2, \dots, N$ . Note that we have dropped the subscript  $k$  in (3.5). We may also rewrite  $Y$  as

$$Y = \text{median} \{W_{(ij)} : j - i < r \text{ and } |D_j - D_i| < t, \\ 1 \leq i \leq j \leq N\} \quad (3.6)$$

where the antirank  $D_i$  is the time position of the  $i$ th smallest data value  $X_{(i)}$ . The GW filter defined by (3.5) has a very interesting structure with the degrees of Winsorization forming a moving mask over the time-rank matrix for any given window of data. In the following we formalize this concept.

For a given input window of size  $N$ , let us define an  $N \times N$  time-rank matrix  $\mathbf{P}$  where the entries  $p_{ij}$  are either 1 or 0. The condition  $p_{ij} = 1$  describes the fact that the  $j$ th input value is  $i$ th smallest in magnitude. Hence, there are exactly  $N$  1's in  $\mathbf{P}$ , with a single value of 1 in each row and column. Such a matrix is called a permutation matrix. If ties occur in the observations, they may be broken with a random assignment. The behavior of the GW filter can be described as follows.

Let  $X_j = X_{(i)}$ , so that  $D_i = j$ . Define a rectangular mask  $M_j$  to contain the set of indexes

$$M_j = \{(m, k) : 1 \leq m, k \leq N, \quad i - r < m < i + r, \\ j \leq k < j + t\}. \quad (3.7)$$

Now define the set  $S_j$  as

$$S_j = \{(m, k) : p_{mk} = 1, \quad (m, k) \in M_j\}. \quad (3.8)$$

Thus,  $S_j$  contains all time-rank combinations which fall in the mask  $M_j$  of temporal size  $t$  and rank size  $2(r-1) + 1$ . In other words, the set  $S_j$  identifies all data values  $X_k$  which are adjacent to  $X_j$  in both time and rank. We then form pairwise averages of  $X_j$  with all  $X_k$  for which  $(m, k)$  is in  $S_j$ . Repeating this for each  $j$ , we finally compute the

output as the median of all these pairwise averages:

$$Y = \text{median} \left\{ \bigcup_{j=1}^N \{W_{jk} : (m, k) \in S_j\} \right\}. \quad (3.9)$$

We will write  $\text{GW}(t, r)$  to denote the GW filter with time-Winsorization parameter  $t$  and rank-Winsorization parameter  $r$ .

Intuitively, if there is an edge or an impulse present in the input sequence, the corresponding data values will occupy higher positions in  $\mathbf{P}$  in terms of ranks. Hence, by judiciously selecting the filter parameters  $t$  and  $r$ , the GW filter may preserve edges and fine details and remove impulses.

As an example, consider an input window  $\{2, 7, 4, -1, 15\}$ . Here we have  $N = 5$ , and we take  $t = 3$  and  $r = 3$ . The time-rank matrix  $\mathbf{P}$  is shown in Fig. 4 with different masks, and the output  $Y$  is given by the median of the set  $\{2, 4.5, 3, 7, 5.5, 4, 1.5, 9.5, -1, 15\}$  which is equal to 4.25.

The  $\text{GW}(t, r)$  filter has the following properties of interest.

*Property 1:* The GW filter is a scale and translation preserving filter; that is, if we denote the output sequence  $\{y_j\}$  of a GW filter for an input sequence  $\{x_j\}$  as  $\{y_j\} = f(\{x_j\})$ , we have

$$f(a\{x_j\} + b\{1\}) = f(\{ax_j + b\}) = af(\{x_j\}) + b\{1\} \quad (3.10)$$

where  $\{1\}$  is the sequence of constant value of 1, and  $a$  and  $b$  are any real constants. This property is a common characteristic of several generalized median filters [4].

*Property 2:* For a linearly increasing (or decreasing) input sequence, that is, for  $x_j = j, j = 0, 1, 2, \dots$ , we have  $y_j = j$ . Note that for any monotonic input sequence, the  $\mathbf{P}$  matrix is diagonal (for increasing sequences) or anti-diagonal (for decreasing sequences). In this case, the GW filter reduces to the rank(time)-Winsorized Wilcoxon filter with  $\min(t, r)$  as the degree of rank(time)-Winsorization.

*Property 3:* The GW filter reduces to the median filter if  $\min(t, r) = 1$  and the Wilcoxon filter if  $(t, r) = (N, N)$ .

*Property 4:* The GW filter reduces to a time-Winsorized Wilcoxon filter if  $r = N$  and to a rank-Winsorized Wilcoxon filter if  $t = N$ .

These properties follow directly from the definition of the GW filter given in (3.5).

In view of properties 3 and 4, we may say that, in general, the degrees of temporal and rank Winsorization control the extent of low-pass action of the GW filter. That is, larger values of  $t$  and  $r$  lead to more averaging being computed before the final median operation is applied. The basic property of impulse noise rejection is provided by this median operation, whereas the Winsorizations in the time and rank domains provide the detail and edge retention characteristics with Gaussian noise smoothing.

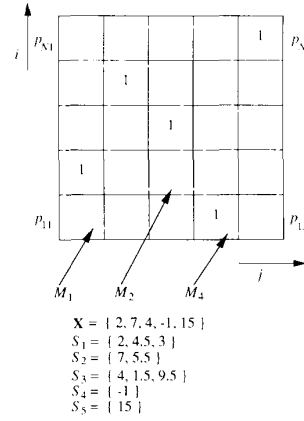


Fig. 4. A typical time-rank matrix with masks for  $N = 5$  and  $(t, r) = (3, 3)$ .

### C. Generalized Wilcoxon filter of Order $m$

So far we have focused our attention on filters that are modifications of Wilcoxon-type filters; that is, the filter output is defined as the median of some pairwise averages of the input data within a window. An obvious modification of the GW filter is to replace the Walsh averages by more general  $m$ -wise averages of the data values that are time-rank adjacent. Modifying (3.5) we get

$$Y = \text{median} \left\{ \frac{1}{m} \sum_{s=1}^m X_{q_s} : |Rq_j - Rq_i| < r, \right. \\ \left. q_j - q_i < t, \quad 1 \leq i < j \leq m, \quad \mathbf{q} \in V_m \right\} \quad (3.11)$$

where  $\mathbf{q} = (q_1, q_2, \dots, q_m)$ ,  $1 \leq m \leq N$  and

$$V_m = \left\{ (q_1, q_2, \dots, q_m) : 1 \leq q_1 \leq q_2 \leq \dots \leq q_m \leq N \right\}. \quad (3.12)$$

The parameter  $m$  will be called the *order* of the filter. We will write  $\text{GW}_m(t, r)$  to mean the generalized Wilcoxon filter of order  $m$ . Obviously,  $\text{GW}_m(t, r)$  reduces to the  $\text{GW}(t, r)$  filter of (3.5) for  $m = 2$ .

With a more general averaging of  $m$  terms, the  $\text{GW}_m$  filter can now be designed to include more linear averaging, which is optimum for Gaussian noise reduction. It is, however, impossible to smooth Gaussian noise optimally and to reject impulses simultaneously. The design values of  $m, t$ , and  $r$  may be obtained analytically to optimize performance according to certain criterion such as the mean square error (MSE); we shall not do so here.

Instead of the set of subscripts  $V_m = \{\mathbf{q} : 1 \leq q_1 \leq q_2 \leq \dots \leq q_m \leq N\}$  in (3.12), if  $\tilde{V}_m = \{\mathbf{q} : 1 \leq q_1 < q_2 < \dots < q_m \leq N\}$  is used, the computational burden will be reduced. Thus, if computation time is a major problem, we can use the filter with  $\tilde{V}_m$  for the subscripts  $\mathbf{q}$  in (3.11) at the expense of slight performance degradation.

Properties 1 and 2 of the GW filter still remain valid for the  $GW_m(t, r)$  filters. Other properties require some additional modifications and follow directly from the definition of the  $GW_m$  filter given by (3.11).

*Property 5:* If  $\min(m, t, r) = 1$ , the  $GW_m$  filter reduces to the median filter, and if  $(m, t, r) = (2, N, N)$ , it reduces to the Wilcoxon filter.

*Property 6:* If  $(m, t, r) = (N, N, N)$ , the  $GW_m$  filter is almost the running mean filter. (If integer  $m$ -tuples  $(q_1, q_2, \dots, q_m)$  are allowed in the set  $\tilde{V}_m$  of (3.12), we would have exactly the running mean filter under this condition.)

Two special cases of interest are obtained when  $r = N$  or  $t = N$ . These yield the  $m$ th-order time-Winsorized and  $m$ th-order rank-Winsorized Wilcoxon filters, respectively. In other words, the filter output of the  $m$ th-order time-Winsorized Wilcoxon filter (or equivalently,  $GW_m(t, N)$  filter) is given by

$$Y = \text{median} \left\{ \frac{1}{m} \sum_{s=1}^m X_{q_s}; q_j - q_i < t, \right. \\ \left. 1 \leq i < j \leq m, \quad \mathbf{q} \in V_m \right\}. \quad (3.13)$$

Similarly, for an  $m$ th-order rank-Winsorized Wilcoxon filter (or  $GW_m(N, r)$  filter), we have

$$Y = \text{median} \left\{ \frac{1}{m} \sum_{s=1}^m X_{(q_s)}; q_j - q_i < r, \right. \\ \left. 1 \leq i < j \leq m, \quad \mathbf{q} \in V_m \right\}. \quad (3.14)$$

In Section IV we show that the  $m$ th-order time-Winsorized Wilcoxon filter has a better overall performance than the rank-Winsorized Wilcoxon filter for filtering certain types of signals of interest. Note, however, that both filters perform identically for monotonic input sequences.

The  $m$ th-order time-Winsorized Wilcoxon filter given by (3.13) can also be defined from an estimate of location. For instance, extending  $P(\theta)$  of (3.1) for  $m = 3$ , we get

$$P(\theta) = \sum_{i=1}^N \sum_{j=1}^N \sum_{\substack{k=1 \\ i \leq j \leq k}}^N (\bar{a}_{j-i+1} - \bar{a}_{j-i})(\bar{a}_{k-i+1} - \bar{a}_{k-i}) \\ \cdot (\bar{a}_{k-j+1} - \bar{a}_{k-j})u(W_{ijk} - \theta) \quad (3.15)$$

where

$$W_{ijk} = \frac{X_i + X_j + X_k}{3}. \quad (3.16)$$

It can be shown by slightly modifying the proof in Appendix A that  $P(\theta)$  is symmetrically distributed, and that

$$E[P(\theta)] = \sum_{i=1}^N \sum_{j=1}^N \sum_{\substack{k=1 \\ i \leq j \leq k}}^N (\bar{a}_{j-i+1} - \bar{a}_{j-i})(\bar{a}_{k-i+1} - \bar{a}_{k-i}) \\ \cdot (\bar{a}_{k-j+1} - \bar{a}_{k-j})/2. \quad (3.17)$$

Hence, for a linearly Winsorized temporal score function, the resulting estimate of location is given by (3.13) with  $m = 3$ . The expression for  $P(\theta)$  in (3.15) can easily be extended for  $m > 3$ . The rank-Winsorized Wilcoxon filter of order  $m$  given by (3.14) can also be defined in a similar manner.

Fig. 5 illustrates a structure for a  $GW_m(t, N)$  filter of order  $m = 3$  and degree  $t = 2$ . Similar structures for filters with different values of  $t$  and  $m$  and for filters with  $\tilde{V}_m$  in place of  $V_m$  in (3.12) can easily be found and are quite similar to that of Fig. 5. Schemes for efficient implementation of the median operation have been devised by other investigators [14], which may be used for finding the median in Fig. 5.

*Computational Complexity:* The number of terms,  $M_c$ , over which the median is computed in a  $GW_m(t, r)$  filter is bounded from above by the number  $N_c$  required in the  $GW_m(d, N)$  filter with  $d = \min(t, r)$ . Hence, in the following analysis, we use the  $GW_m(d, N)$  filter as our yardstick for the computational complexity of the  $GW_m(t, r)$  filter. It is shown in Appendix B that

$$N_c = \sum_{i=1}^N {}_k H_{m-1} \quad (3.18)$$

where  $k = \min(d, N - i + 1)$  and  ${}_k H_j$  is the number of combinations, allowing duplication, of  $j$  elements out of  $k$  possible elements and is equal to  $(k + j - 1)!/j!(k - 1)!$ . When  $\tilde{V}_m$  is used for  $\mathbf{q}$  in (3.11), this number becomes

$$\tilde{N}_c = \sum_{i=1}^{N-m+1} \tilde{k} C_{m-1} \quad (3.19)$$

where  $\tilde{k} = \min(d - 1, N - i)$  and  ${}_k C_j = k!/j!(k - j)!$ . Table I shows values of  $N_c$  and  $\tilde{N}_c$  for various values of  $N$ ,  $d$ , and  $m$ . As can be seen from Table I,  $N_c$  is considerably larger than  $N$ , implying that the  $GW_m(d, N)$  filter requires more computation to get an output point than does the median filter for the same window size. This computational requirement can, however, be reduced by exploiting the fact that only a few averages have to be updated for each output point computation, and by using  $\tilde{V}_m$  in (3.11) instead of  $V_m$ .

#### D. A Dual Class of Filters

It is noteworthy that the filters we have considered in the previous section are versions of a more general class of filters whose filter transformation  $T_1(\mathbf{X})$  can be expressed as

$$T_1(\mathbf{X}) = \text{med} \{ \mathbf{f}(\mathbf{X}) \} \quad (3.20)$$

where  $\mathbf{f}$  is a vector linear operation mapping  $\mathcal{R}^N$  to  $\mathcal{R}^P$  with  $P \leq N$ , the median operator is a nonlinear function mapping  $\mathcal{R}^P$  to  $\mathcal{R}$ , and  $\mathbf{X}$  is the vector of  $N$  observations in a window of size  $N$ . With  $\mathbf{f}$  the averaging operation and  $P = 1$ , we get the running mean filter; with  $\mathbf{f}$  the identity function,  $T_1$  represents the median filter; other interesting

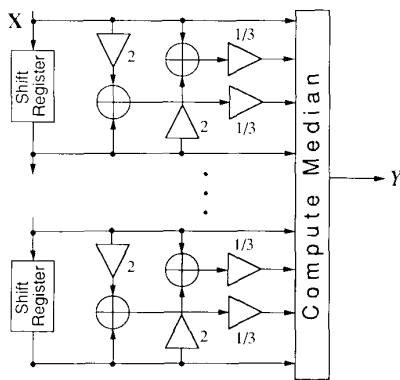


Fig. 5. A structure for the  $GW_3(2, N)$  filter.

TABLE I  
VALUES OF  $N_e$  AND  $\tilde{N}_e$  FOR DIFFERENT VALUES OF  $N$ ,  $d$ , AND  $m$

$(N,m,d)$	$N_e$	$\tilde{N}_e$	$(N,m,d)$	$N_e$	$\tilde{N}_e$	$(N,m,d)$	$N_e$	$\tilde{N}_e$
(5,2,2)	9	4	(15,2,2)	29	14	(25,2,2)	49	24
(5,2,3)	12	7	(15,2,3)	42	27	(25,2,3)	72	47
(5,3,2)	13	*	(15,2,4)	54	39	(25,2,4)	94	69
(5,3,3)	22	3	(15,3,2)	43	*	(25,3,2)	73	*
(5,2,5)	15	10	(15,3,3)	82	13	(25,3,3)	142	23
(7,2,2)	13	6	(15,3,4)	130	37	(25,3,4)	230	67
(7,2,3)	18	11	(15,4,2)	57	*	(25,4,2)	97	*
(7,3,2)	19	*	(15,4,3)	135	*	(25,4,3)	235	*
(7,3,3)	34	5	(15,4,4)	255	12	(25,4,4)	455	22
(7,2,7)	28	21	(15,2,15)	120	105	(25,2,25)	325	300

\* not defined

filters such as the linear-median hybrid (LMH) filters [6], [9] may be generated from this class of nonlinear filters. In particular, the LMH filter whose output is the median of means of disjoint subsets of the observations in a window can naturally be considered as a special case of this class, when the  $P$  components of  $f(\mathbf{X})$  are averages of contiguous observations in disjoint subsets of  $\mathbf{X}$ . A primary difference between the LMH filter and the  $GW_m$  filter lies in the way in which subsets of observations are used in the linear operation in (3.20). (In the  $GW_m$  filter, all the subsets are of the same size and they are not disjoint, whereas they are generally of different size and composed of disjoint subsets of contiguous observations in the LMH filter.) It is clear that the choice of  $f$  in (3.20) which yields the Wilcoxon filter is  $f(\mathbf{X}) = \{(X_i + X_j)/2, 1 \leq i \leq j \leq N\}$ .

Let us now consider a dual class of nonlinear filters which can be constructed by an interchange of the linear and nonlinear operations in (3.20); the resulting class of nonlinear filters is defined by the transformations

$$T_2(\mathbf{X}) = \text{mean} \{g(\mathbf{X})\} \quad (3.21)$$

where  $g$  is generally a nonlinear vector function mapping  $\mathbb{R}^N$  to  $\mathbb{R}^P$ . The  $P$  components of  $g(\mathbf{X})$  may, in many cases of interest, be interpreted as each arising from some non-

linear transformation of a subset of the  $N$  observations. If  $g$  is the full median operation and  $P = 1$ , then  $T_2$  represents the median filter; if  $g$  is the identity function, then  $T_2$  represents the running mean filter. By choosing other functions or operations for  $g$ , more interesting filters such as the  $L$ -filters and the KNN filters can be derived from this general class of filters. It should also be noted that the nonlinear mean filters, which have recently been considered and analyzed in [7], are of this type. A particularly interesting nonlinear filter class giving filters dual to the  $GW_m(N, N)$  filters is obtained with each of the  $P$  components of  $g(\mathbf{X})$  being median  $\{X_{q_1}, X_{q_2}, \dots, X_{q_m}\}$  for  $P = {}_N H_m$  or  $\tilde{P} = {}_N C_m$  possible combinations of subscripts  $(q_1, q_2, \dots, q_m)$  in  $V_m$  or  $\tilde{V}_m$ , respectively. In this case,  $T_2(\mathbf{X})$  represents a class of  $L$ -filters with non-equal symmetric coefficients [25]. More explicitly, the filter transformation becomes, for  $m$  odd and  $\tilde{V}_m$  in the definition of the filter,

$$T_2(\mathbf{X}) = \frac{1}{N C_m} \sum_{i=(m+1)/2}^{N-(m-1)/2} \binom{N-(m-1)/2}{i-1} \binom{N-(m-1)/2}{N-i} X_{(i)} \quad (3.22)$$

This filter transformation becomes, for example,

$$T_2(\mathbf{X}) = \frac{1}{21} \{6X_{(3)} + 9X_{(4)} + 6X_{(5)}\} \quad (3.23)$$

when  $N = 7$  and  $m = 5$ . It can also be shown that the filters with  $m = 2I$  and  $m = 2I - 1$  are exactly the same for given window size  $N$  and any positive integer  $I$  [25]. For a general discussion on  $L$ -filters, see [3] and [4].

Note that the above two dual classes of filters are not disjoint. Any given filter may be considered to belong more naturally to one of these two classes.

#### IV. ANALYSIS AND SIMULATION OF THE $GW_m$ FILTER PERFORMANCE

In this section, the performance of the  $GW_m(t, N)$  filter will be considered for an ideal edge input and inputs with narrow pulses of various widths. Performances of several filters will then be compared through computer simulations. Results are given of computer simulations for two one-dimensional signals containing edges, narrow pulses, and slowly varying portions and additive noise.

##### A. Performance for an Ideal Edge and Narrow Pulses

A general analysis of the performance of the  $GW_m(t, r)$  filter has to be based on probabilistic considerations. However, for  $r = N$ , a simpler deterministic analysis can be given. It also turns out that the  $GW_m(t, N)$  filters appear to offer the best compromise in performance against impulsive noise for edge and detail preservation. Let us consider an input with an ideal edge which can be represented by the sequence  $\{\dots 00001111\dots\}$ . It is quite straightforward to obtain the outputs of different  $GW_m(t, N)$  filters for this input. Some results for window sizes  $N = 5$ ,  $N = 7$ , and  $N = 9$  are given in Table II(a) and (b). More generally, it can be proved (see Appendix

TABLE II  
OUTPUTS OF  $GW_m(t, N)$  FILTERS OF DIFFERENT WINDOW SIZES FOR AN IDEAL EDGE INPUT

(a) $m=2$	
$N=5$	
GW <sub>2</sub> (2,5) Filter	... 0 0 0 0 1 1 1 1 ...
GW <sub>2</sub> (3,5) Filter	... 0 0 0 1/4 3/4 1 1 1 ...
GW <sub>2</sub> (4,5) Filter	... 0 0 0 1/2 1/2 1 1 1 ...
GW <sub>2</sub> (5,5) Filter	... 0 0 0 1/2 1/2 1 1 1 ...
$N=7$	
GW <sub>2</sub> (2,7) Filter	... 0 0 0 0 1 1 1 1 ...
GW <sub>2</sub> (3,7) Filter	... 0 0 0 1/4 3/4 1 1 1 ...
GW <sub>2</sub> (4,7) Filter	... 0 0 0 1/2 1/2 1 1 1 ...
GW <sub>2</sub> (7,7) Filter	... 0 0 0 1/2 1/2 1 1 1 ...
$N=9$	
GW <sub>2</sub> (2,9) Filter	... 0 0 0 1 2 1 1 1 ...
GW <sub>2</sub> (3,9) Filter	... 0 0 0 1/4 3/4 1 1 1 ...
GW <sub>2</sub> (4,9) Filter	... 0 0 0 1/2 1/2 1 1 1 ...
GW <sub>2</sub> (7,9) Filter	... 0 0 1/4 1/2 1/2 3/4 1 1 ...
GW <sub>2</sub> (9,9) Filter	... 0 0 1/2 1/2 1/2 1/2 1 1 ...
(b) $m=3$	
$N=5$	
GW <sub>3</sub> (3,5) Filter	... 0 0 0 1/3 2/3 1 1 1 ...
GW <sub>3</sub> (4,5) Filter	... 0 0 0 1/3 2/3 1 1 1 ...
GW <sub>3</sub> (5,5) Filter	... 0 0 0 1/3 2/3 1 1 1 ...
$N=7$	
GW <sub>3</sub> (3,7) Filter	... 0 0 0 1/3 2/3 1 1 1 ...
GW <sub>3</sub> (4,7) Filter	... 0 0 0 1/3 2/3 1 1 1 ...
GW <sub>3</sub> (5,7) Filter	... 0 0 0 1/3 2/3 1 1 1 ...
GW <sub>3</sub> (6,7) Filter	... 0 0 1/3 1/3 2/3 2/3 1 1 ...
GW <sub>3</sub> (7,7) Filter	... 0 0 1/3 1/3 2/3 2/3 1 1 ...

C) that the  $GW_2(2, N)$  filters of any window size  $N = 2n + 1 > 3$  preserve an ideal edge as does the median filter, while the Wilcoxon filter, which is the  $GW_2(N, N)$  filter, does not preserve ideal edges. In fact, it is possible to obtain a general result on the ideal edge performance of the  $GW_2(t, N)$  filters for any window of size  $N > 3$ . From the discussion in Appendix C, we see that as  $t$  increases for fixed  $N$ , the extent of smearing of the ideal edge remains constant with two values only around the edge being smeared, up to a certain value of  $t$ ; beyond this the edge begins to get smeared more, with more output values of  $1/2$  around the ideal edge position. Although we do not provide analytic performance characterization of  $GW_m(t, N)$  filters with  $m > 2$ , it can be expected that more smearing of edges will occur with increasing  $m$ , as seen in Table II(b).

For narrow pulses of heights 1 and various widths  $w \geq 2$ , the outputs of several filters of window size  $N = 7$  are given in Table III. Table III shows that the  $GW_2(2, N)$  and  $GW_3(3, N)$  filters generally retain narrow pulses better than the median and Wilcoxon filters of the same window size 7. Table III also shows the tradeoff between narrow pulse retention and output pulse integrity; in order to

TABLE III  
OUTPUTS OF  $GW_m(t, N)$  FILTERS FOR NARROW PULSE INPUT

(a) $w=2, N=7$	
Input	... 0 0 0 0 1 1 0 0 0 0 ...
Median Filter	... 0 0 0 0 0 0 0 0 0 0 ...
GW <sub>2</sub> (2,7) Filter	... 0 0 0 0 0 0 0 0 0 0 ...
GW <sub>2</sub> (7,7) Filter	... 0 0 0 0 0 0 0 0 0 0 ...
GW <sub>3</sub> (3,7) Filter	... 0 0 0 0 1/3 1/3 0 0 0 0 ...
(b) $w=3, N=7$	
Input	... 0 0 0 0 1 1 1 0 0 0 0 ...
Median Filter	... 0 0 0 0 0 0 0 0 0 0 ...
GW <sub>2</sub> (2,7) Filter	... 0 0 0 0 1/2 1/2 1/2 0 0 0 0 ...
GW <sub>2</sub> (7,7) Filter	... 0 0 0 1/2 1/2 1/2 1/2 1/2 0 0 0 ...
GW <sub>3</sub> (3,7) Filter	... 0 0 0 0 1/2 2/3 1/2 0 0 0 0 ...
(c) $w=4, N=7$	
Input	... 0 0 0 0 1 1 1 1 0 0 0 0 ...
Median Filter	... 0 0 0 0 1 1 1 1 0 0 0 0 ...
GW <sub>2</sub> (2,7) Filter	... 0 0 0 0 1 1 1 1 0 0 0 0 ...
GW <sub>2</sub> (7,7) Filter	... 0 0 0 1/2 1/2 1/2 1/2 1/2 1/2 0 0 0 ...
GW <sub>3</sub> (3,7) Filter	... 0 0 0 1/3 2/3 2/3 2/3 2/3 1/3 0 0 0 ...

get reasonable narrow pulse retention characteristics, a small amount of smearing and loss of amplitude is unavoidable. If we consider a pulse of width  $w = 1$  (which is an impulsive noise component in practice) for which the input sequence is  $\{\dots 0 0 1 0 0 \dots\}$ , the outputs of the above three filters will be exactly the same, the all-zero sequence  $\{\dots 0 0 0 0 0 \dots\}$ , which implies good impulsive noise rejection properties of the filters. Although the above observations have been made from results for a fixed window size, an analysis of the narrow-pulse retention characteristics of the  $GW_2(2, N)$  filter for any window of size  $N = 2n + 1 > 3$  (see Appendix D) shows that the  $GW_2(2, N)$  filter retains pulses of width  $w \geq n + 1$  and it also retains pulses of width  $w = n$  with half the amplitude, as shown in Table III. It should be noted that the median filter with the same window size will also retain pulses of width  $w \geq n + 1$ , but will remove pulses of width  $w \leq n$ . It is also possible to explain the pulse response of the Wilcoxon filters. For the Wilcoxon filter, the response depends not on the pattern of the ones and zeros for a binary input, but only on the numbers of ones and zeros inside a window. Thus, the ideal edge performance of the Wilcoxon filter can be used to obtain its ideal pulse performance. For example, when  $N = 7$  and  $w = 2$ , the Wilcoxon filter will not be able to retain this pulse because at any time there are at most two ones, thus producing a zero as the output. From its edge performance we conclude that when  $N = 7$  and  $w = 3$ , the output of the Wilcoxon filter will be a smeared and reduced pulse  $\{1/2, 1/2, 1/2, 1/2, 1/2\}$ , and for a pulse of width  $w = 4$ , the output of the Wilcoxon filter

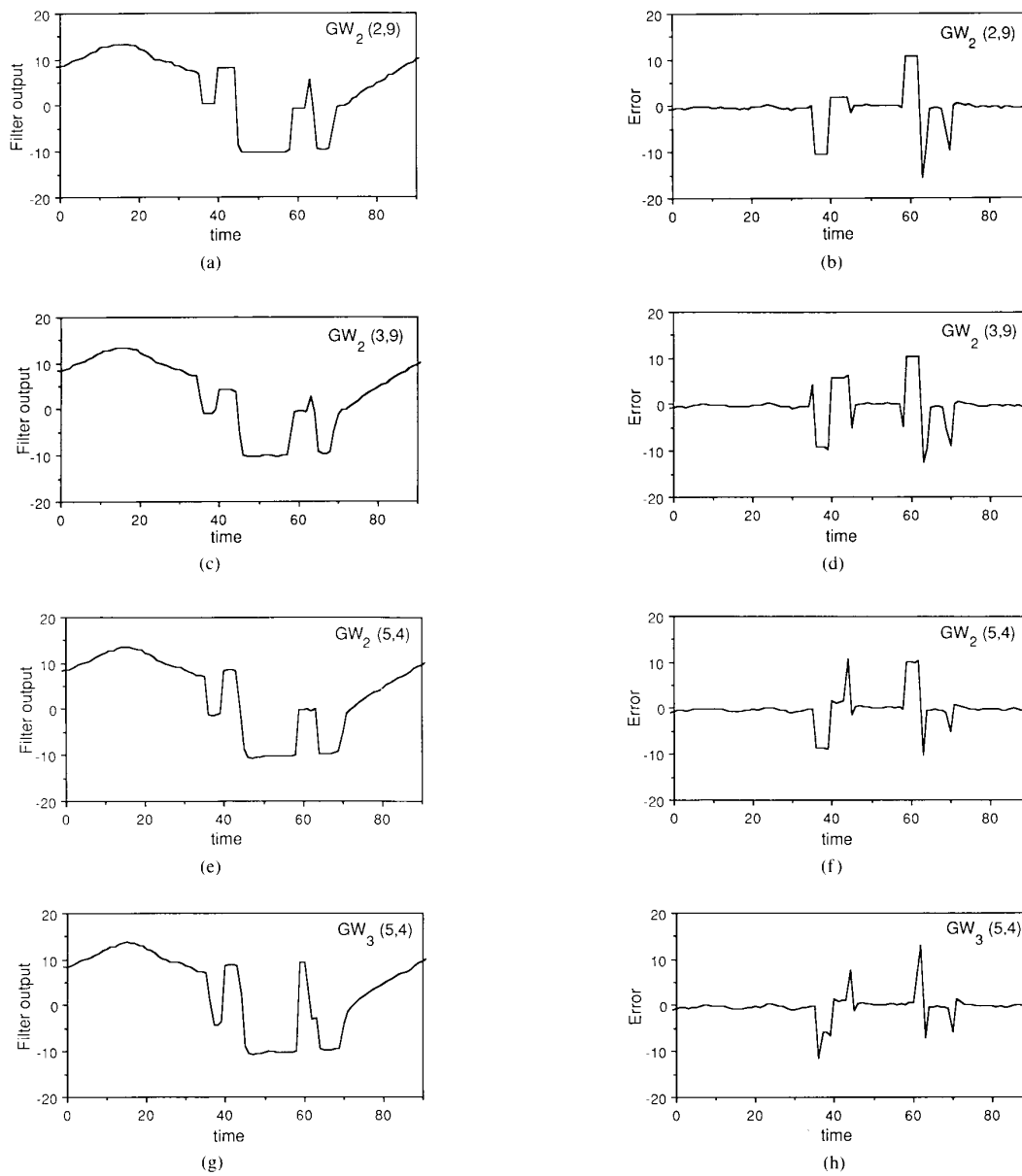


Fig. 6. Simulation results with  $N = 9$ . (a) Output of the  $GW_2(2, 9)$  filter. (b) Error in the output of the  $GW_2(2, 9)$  filter. (c) Output of the  $GW_2(3, 9)$  filter. (d) Error in the output of the  $GW_2(3, 9)$  filter. (e) Output of the  $GW_2(5, 4)$  filter. (f) Error in the output of the  $GW_2(5, 4)$  filter. (g) Output of the  $GW_3(5, 4)$  filter. (h) Error in the output of the  $GW_3(5, 4)$  filter.

with window size 7 will again be a smeared and reduced pulse  $\{1/2, 1/2, 1/2, 1/2, 1/2, 1/2\}$ ; this is in agreement with the results in Table III.

### B. Simulation Results for Noisy Signals

*Example 1 (Effect of Winsorization):* In the first example, we consider the signal of Fig. 3(b) as an input to the  $GW_m$  filters with  $N = 9$ , with outputs and errors shown

in the various parts of Fig. 6. Fig. 6(a) and (c) shows outputs produced by the  $GW_2(2, 9)$  and  $GW_2(3, 9)$  filters, which are the counterparts of the rank-Winsorized Wilcoxon filters producing the results of Fig. 3(e) and (f), respectively. Fig. 6(b) and (d) shows the deviations of these outputs from the original signal. Note that the time-Winsorized filters with  $t = 2$  and 3, respectively, are better compared to the rank-Winsorized Wilcoxon filters with

the small degree of Winsorization used here. Two other  $GW_m$  filter outputs with  $(m, t, r) = (2, 5, 4)$  and  $(m, t, r) = (3, 5, 4)$  are shown in Fig. 6(e) and (g), respectively. These last two filters are better able to retain the general shape of the signal and to recover both pulses and reject the impulse effectively; the deviations of these outputs from the original are shown in Fig. 6(f) and (h).

*Example 2 (Noise Suppression and Edge and Pulse Preservation):* One set of simulations was run to illustrate the performance at edges for narrow pulses, for four filters of window size  $N = 7$ . These were the median, the  $GW_2(7, 7)$  (Wilcoxon), the  $GW_2(2, 7)$ , and the  $GW_3(4, 6)$ . An edge of height 10 followed by a negative pulse of amplitude 10 and width 3 was the signal component, to which i.i.d. Gaussian noise of mean zero and variance  $\sigma^2 = 1$  was added. The outputs, averaged over independent 100 runs, are shown in Fig. 7. It is clear that the  $GW_2(2, 7)$  filter has good edge retention characteristics as explained from the analysis given above. The  $GW_3(4, 6)$  filter smears the edge more but its pulse retention capability is better than that of the Wilcoxon or median filters.

In Table IV we show the output mean square errors for filters with a constant level signal with additive i.i.d. Gaussian zero-mean, unit variance noise, obtained as time-averages for a 2000-point input sequence. In the case of the median filters, the MSE's agree with those reported in earlier work [26], [27]. It can be observed from Table IV that the Gaussian noise suppression capability of the  $GW_2(2, 5)$  filter is better than that of the median filter of window size 3 and 5, and is somewhat less than that of the window-size 7 median filter.

*Example 3 (Detail Retention):* To illustrate the ability of the  $GW_m$  filters to retain details, let us consider the signal of Fig. 8(a) which has 5 narrow pulses on a slowly varying background; Fig. 8(b) shows the noisy signal to be filtered, with additive zero-mean white Gaussian noise of standard deviation 2.5 and 5 impulses. This noisy signal was processed using different filtering schemes with different window sizes. The window-size 3 median filter [med(3)] output of Fig. 8(c) shows all 5 narrow signal pulses, but the output is quite noisy. Fig. 8(d) shows the deviation of the med(3) filter output from the original signal. If a window-size 5 median filter [med(5)] is used, the narrower signal pulse is obliterated. On the other hand, the  $GW_2(2, 5)$  filter of window size 5 retains all five pulses [Fig. 8(e)] and gives better noise suppression. The deviation from the original of the  $GW_2(2, 5)$  filter output is given in Fig. 8(f). In order to better appreciate the Gaussian noise smoothing properties of the  $GW_2(2, 5)$  filter compared to those of the median filters (of window sizes 3 and 5), we have computed the output mean square errors (MSE) of these filters. In determining the MSE, we only consider those intervals where the original signal has no pulses present. The MSE for the med(3) filter is 2.82, whereas it is 1.66 for the med(5) filter and is 1.31 for the

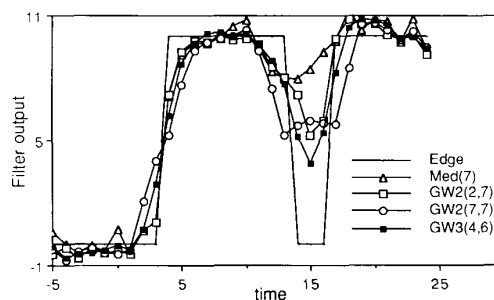


Fig. 7. Simulation results of edge performance and pulse retention of some  $GW_m$  filters with  $N = 7$ .

TABLE IV  
MEAN SQUARE ERROR SIMULATION OF THE  $GW_m$  FILTERS FOR A CONSTANT SIGNAL IN NOISE FOR DIFFERENT WINDOW SIZES

$N$	3	5	7
Filter			
$GW_2(1,1)$	0.448	0.290	0.214
$GW_2(2,N)$	-	0.255	0.182
$GW_2(N,N)$	-	0.221	0.162
$GW_2(4,6)$	-	-	0.177
$GW_3(4,6)$	-	-	0.176

$GW_2(2, 5)$  filter. For comparison, the output of the Wilcoxon filter of the same window size (i.e., the  $GW_2(5, 5)$  filter of window size 5) and the deviation in its output from the original are shown in Fig. 8(g) and (h), respectively. The widening and smoothing of the pulses is evident there.

For more smoothing, larger window sizes are needed. The window-size 7 median filter output is shown in Fig. 8(i), and the deviation of the output from the original is shown in Fig. 8(j). It shows clearly only two of the five signal pulses. The  $GW_2(2, 7)$  filter output shown in Fig. 8(k) is clearly better in that more signal pulses are retained. The output of the  $GW_2(3, 5)$  filter of window size 7 is shown in Fig. 8(l). This performance is almost the same as that of the  $GW_2(2, 7)$  filter, which uses more severe temporal Winsorization but no rank Winsorization. The  $GW_2(4, 6)$  filter output shown in Fig. 8(m) is interesting because all five pulses are present there. This filter represents somewhat less severe Winsorization than the  $GW_2(3, 5)$  filter. Finally, the  $GW_3(4, 6)$  filter (of window size 7 also) shows more smoothing and rounding-off of the pulses, which are nonetheless all preserved in the output shown in Fig. 8(n). In general, the distortion of pulse shape is more severe if more Gaussian noise smoothing is desired.

We should note, however, that the median filter will be able to recover more pulses if smaller window sizes ( $N = 3$ , for example) are used; the Gaussian noise smoothing will generally become worse. Also, the DW-MTM filter [4] has better edge preserving and Gaussian noise smoothing characteristics compared to the  $GW_m$  filter. The



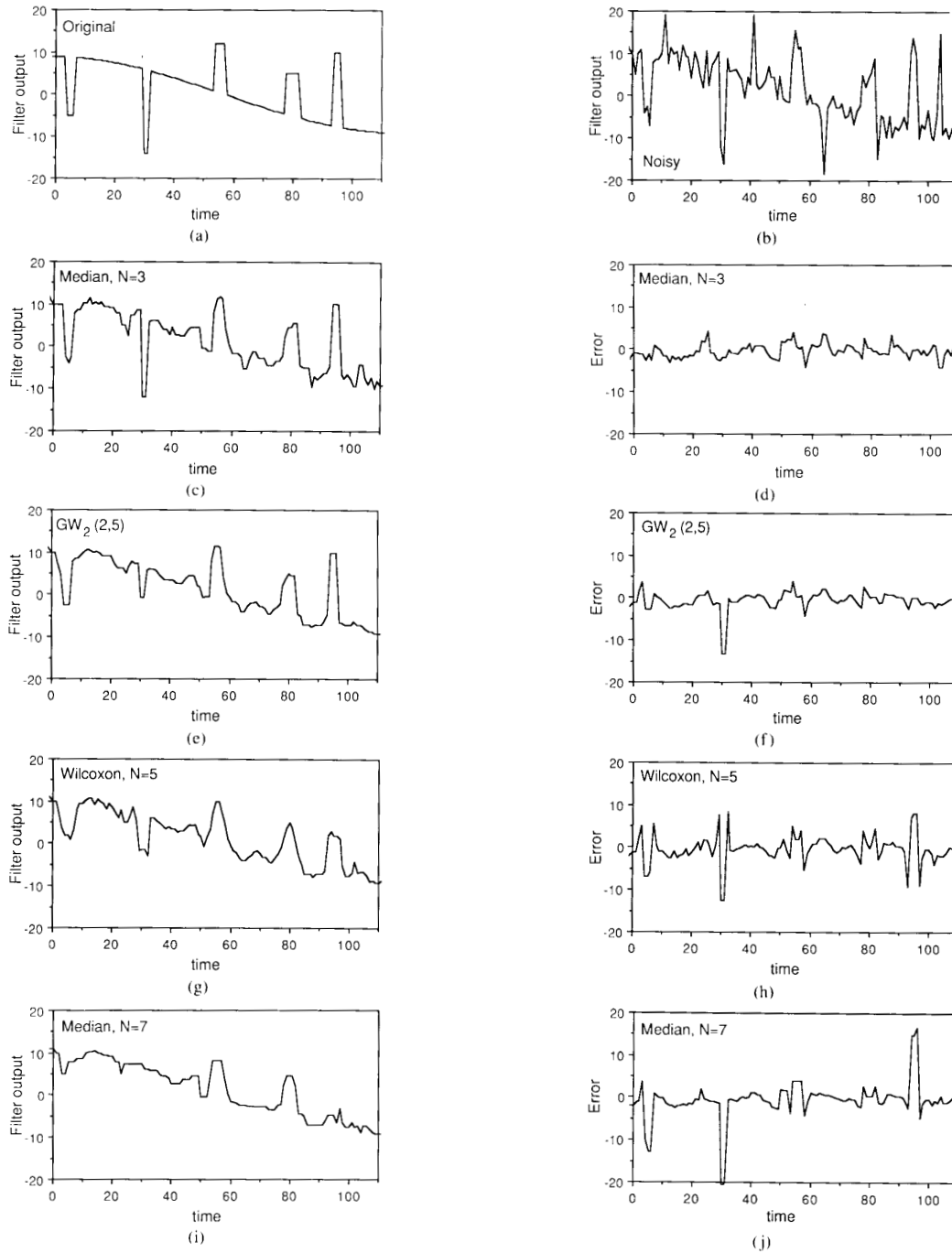


Fig. 8. Simulation results. (a) Original signal. (b) Noisy signal. (c) Output of the median filter with  $N = 3$ . (d) Error in the output of the median filter with  $N = 3$ . (e) Output of the  $GW_2(2, 5)$  filter with  $N = 5$ . (f) Error in the output of the  $GW_2(2, 5)$  filter with  $N = 5$ . (g) Output of the Wilcoxon filter with  $N = 5$ . (h) Error in the output of the Wilcoxon filter with  $N = 5$ . (i) Output of the median filter with  $N = 7$ . (j) Error in the output of the median filter with  $N = 7$ .

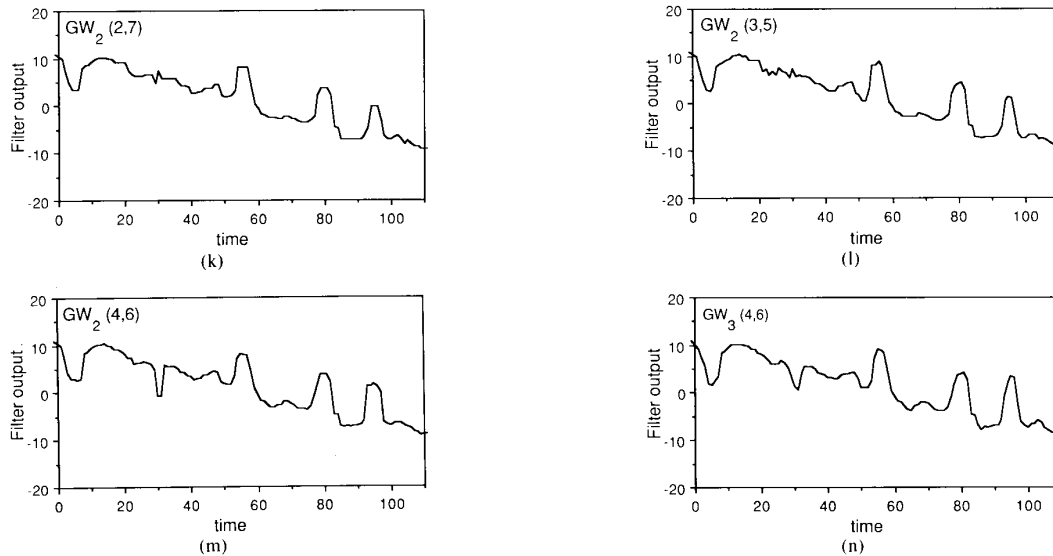


Fig. 8. (Continued.) (k) Output of the  $GW_2(2, 7)$  filter with  $N = 7$ . (l) Output of the  $GW_2(3, 5)$  filter with  $N = 7$ . (m) Output of the  $GW_2(4, 6)$  filter with  $N = 7$ . (n) Output of the  $GW_3(4, 6)$  filter with  $N = 7$ .

output of the DW-MTM filter is the average of observations inside its window which fall within an interval  $[m_k - q, m_k + q]$  where  $m_k$  is the sample median of the observation in a smaller window centered at  $k$ , with  $q$  being a parameter. The DW-MTM filter, however, requires explicit information about the original signal and noise statistics to set the value of  $q$  appropriately (for example, the minimum edge height and the variance of the Gaussian noise process) in order to have better performance than the  $GW_m$  filter.

## V. SUMMARY

In this paper we have examined a class of filters, the  $R$ -filters, that have a basis in the class of rank estimates ( $R$ -estimates) of statistical theory. This class of filters includes the common filters such as the median and the Wilcoxon filters. The investigation of these  $R$ -filters complements recent work on nonlinear filters based on the classes of  $L$ - and  $M$ -estimates of robustness theory. We introduced here specifically the generalized Wilcoxon filters of order  $m$  which incorporate both the temporal and rank information of the input sequence within a finite moving window to compute the output. It was shown that with proper choice of order of averaging  $m$  and degrees of Winsorization, the  $GW_m$  filter can be designed to perform quite satisfactorily for filtering of signals containing edges and details as well as additive white Gaussian and impulsive noise components.

## APPENDIX A

### PROOF OF SYMMETRY OF $T(\theta)$ AND $P(\theta)$

In this appendix we show that  $T(\theta)$  of (2.6) and  $P(\theta)$  of (3.1) are symmetrically distributed random variables. We will assume that  $\mathbf{X} = (X_1, X_2, \dots, X_N)$  is a vector

of  $N$  i.i.d. continuous random variables and  $-\mathbf{X} = (-X_1, -X_2, \dots, -X_N)$ .

*Lemma:* Suppose  $X_1, X_2, \dots, X_N$  are symmetrically distributed about 0. For a function  $g: \mathcal{R}^N \rightarrow \mathcal{R}$ , if  $g(\mathbf{X}) + g(-\mathbf{X}) = \mu$ , then  $g(\mathbf{X})$  is symmetrically distributed about  $\mu/2$ .

*Proof:* Since  $\mathbf{X}$  and  $-\mathbf{X}$  have the same distribution, it is clear that  $g(\mathbf{X})$  and  $g(-\mathbf{X})$  are also identically distributed. That is, we have  $P[g(\mathbf{X}) \leq y] = P[g(-\mathbf{X}) \leq y]$ . Then the lemma follows trivially since  $P[g(\mathbf{X}) \leq \mu/2 - y] = P[g(\mathbf{X}) \geq \mu/2 + y]$ .

*Proposition:* If  $X_1, X_2, \dots, X_N$  are symmetrically distributed about  $\theta$ , then  $T(\theta)$  of (2.6) and  $P(\theta)$  of (3.1) are symmetrically distributed about the points of symmetry  $\sum_{i=1}^N a_i/2$  and  $\sum_{i=1}^N \bar{a}_i/2$ , respectively.

*Proof:* Without loss of generality, let  $\theta = 0$  [we can always do this by defining a new vector of random variables  $(X_1 - \theta, X_2 - \theta, \dots, X_N - \theta)$ ]. From (2.6) with  $g(\mathbf{X}) = T(0)$ , we have

$$\begin{aligned}
 g(\mathbf{X}) + g(-\mathbf{X}) &= \sum_{i=1}^N \sum_{j=1}^N (a_{j-i+1} - a_{j-i}) \\
 &\quad \cdot \{u(W_{(ij)}) + u(-W_{(n-j+1, n-i+1)})\} \\
 &= \sum_{i=1}^N \sum_{j=1}^N (a_{j-i+1} - a_{j-i}) \{u(W_{(ij)}) + u(-W_{(ij)})\} \\
 &= \sum_{i=1}^N \sum_{j=1}^N (a_{j-i+1} - a_{j-i}) \\
 &= \sum_{i=1}^N a_i.
 \end{aligned} \tag{A1.1}$$

Hence, from the lemma, we have  $E[T(\theta)] = \sum_{i=1}^N a_i/2$ . The proof for  $P(\theta)$  is similar to the one given above.

APPENDIX B  
CALCULATION OF  $N_e$  AND  $\tilde{N}_e$

When the set  $V_m$  is used, the number of  $m$ -tuples in  $V_m$ , given that  $q_1 = i$ , is  ${}_d H_{m-1}$  for  $d \leq N - i + 1$  and  ${}_{N-i+1} H_{m-1}$  for  $d > N - i + 1$  where  ${}_k H_j$  is the number of combinations, with duplication allowed, of  $j$  items out of  $k$  types and is equal to  $(k + j - 1)!/j!(k - 1)!$ . Now summing these values for  $i = 1, 2, \dots, N$ , we obtain

$$N_e = \sum_{i=1}^N {}_k H_{m-1} \tag{A2.1}$$

where  $k = \min(d, N + 1 - i)$ . For example, if  $N = 5$ ,  $m = 3$ , and  $d = 4$ , the possible set of three indexes are  $\{(111), (112), (113), (114), (122), (123), (124), (133), (134), (144)\}$  when  $i = 1$ ,  $\{(222), (223), (224), (225), (233), (234), (235), (244), (245), (255)\}$  when  $i = 2$ ,  $\{(333), (334), (335), (344), (345), (355)\}$  when  $i = 3$ ,  $\{(444), (445), (455)\}$  when  $i = 4$ , and  $\{(555)\}$  when  $i = 5$ .

If the set  $\tilde{V}_m$  is used for the subscripts ( $q_1, q_2, \dots, q_m$ ), the number of  $m$ -tuples in  $\tilde{V}_m$  given that  $q_1 = i$  is  ${}_{d-1} C_{m-1}$  for  $d \leq N - i$  and  ${}_{N-i} C_{m-1}$  for  $d > N - i$ . Summing these values for  $i = 1, 2, \dots, N - m + 1$ , we obtain

$$\tilde{N}_e = \sum_{i=1}^{N-m+1} \bar{k} C_{m-1} \tag{A2.2}$$

where  $\bar{k} = \min(d - 1, N - i)$  and  ${}_k C_j = k!/(k - j)!j!$ . Here the upper limit of the summation is  $N - m + 1$ , since each index is allowed to be chosen only once. For the same values of  $m, d$ , and  $N$  as in the example above, the possible set of three indexes are now  $\{(123), (124), (134)\}$  when  $i = 1$ ,  $\{(234), (235), (245)\}$  when  $i = 2$ , and  $\{(345)\}$  when  $i = 3$ .

APPENDIX C  
AN ANALYSIS OF IDEAL EDGE PERFORMANCE OF THE  
GW<sub>2</sub>( $t, N$ ) FILTERS

In Appendixes C and D, we will use the notation  $x^k$  to denote the set  $\{x_{k-n}, \dots, x_k, \dots, x_{k+n}\}$ , the data in a window centered at time  $k$ . We will also denote by  $z^k$  the set  $\{(x_i + x_j)/2, k - n \leq i \leq j \leq k + n, j - i < t\}$ . Now consider Fig. 9. A dot in the  $p$ th column and  $q$ th row represents an element in  $z^k$  with value  $(x_{k-n+p-1} + x_{k-n+q-1})/2$ , and the total number of dots is  $N_e = (2N + 1 - t)t/2$ . If there are  $J$  contiguous zeros and  $N - J$  contiguous ones in  $x^k$ , we can easily see that each element of  $z^k$  represented by a dot located to the left of and above the solid lines is a zero; any value represented by a dot located to the right of and below the solid lines is a 1; and all other elements of  $z^k$  have value  $1/2$ . After a simple enumeration, we obtain the number of zeros  $N(0)$  and the

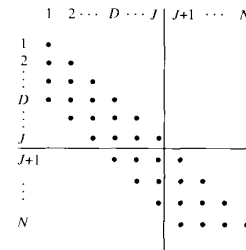


Fig. 9. A representation of the pairs of indexes in computation of the GW<sub>2</sub>( $t, N$ ) filter outputs.

number of ones  $N(1)$  as follows:

- 1) if  $J \leq t$ ,  $N(0) = J(J + 1)/2$  and  $N(1) = t(t + 1)/2 + (N - J - t)t$ ;
- 2) if  $t + 1 \leq J \leq N - t$ ,  $N(0) = t(t + 1)/2 + (J - t)t$  and  $N(1) = t(t + 1)/2 + (N - J - t)t$ ; and
- 3) if  $J \geq N - t + 1$ ,  $N(0) = t(t + 1)/2 + (J - t)t$  and  $N(1) = (N - J)(N - J + 1)/2$ .

From these numbers, we have the following values of outputs for the GW<sub>2</sub>( $t, N$ ) filter, assuming  $N_e = (2N + 1 - t)t/2$  is an odd number:

$$y_k = \begin{cases} 0 & \text{if } J \in B_0 \\ 1 & \text{if } J \in B_1 \\ 1/2 & \text{otherwise} \end{cases} \tag{A3.1}$$

where  $J$  is the number of zeros in the window centered at time  $k$ ,

$$B_0 = \left\{ J : J \leq t, J(J + 1) \geq N_e + 1 \right\} \cup \left\{ J : J \geq t + 1, J \geq \frac{N_e + 1}{2t} - \frac{t + 1}{2} + t \right\} \tag{A3.2}$$

and

$$B_1 = \left\{ J : (N - J) \leq t - 1, (N - J) \cdot (N - J + 1) \geq N_e + 1 \right\} \cup \left\{ J : (N - J) \geq t, (N - J) \geq \frac{N_e + 1}{2t} - \frac{t + 1}{2} + t \right\}. \tag{A3.3}$$

Note that  $B_1$  is almost the same as  $B_0$  with  $(N - J)$  replacing  $J$ ; this implies a symmetric behavior around the edge of the GW<sub>2</sub>( $t, N$ ) filter. Even though we have considered only the case when  $N_e$  is an odd number, an almost identical development can be used for the case when  $N_e$  is an even number.

As a specific example of the above result, let us consider the case  $t = 2$ . Let  $N = 2n + 1$  be the window size, and let the sequence of the ideal edge input be  $\{\dots 0 0 0 0 1 1 1 1 \dots\}$  with the last zero and the first one oc-

curing at  $k = I$  and at  $k = I + 1$ , respectively. Since  $N_e = 2N - 1$  is an odd number, we can directly use the above result to conclude that

$$y_k = \begin{cases} 0 & \text{if } k \leq I \\ 1 & \text{if } k \geq I + 1 \end{cases} \quad (\text{A3.4})$$

because we can simplify (A3.2) and (A3.3) to get

$$B_0 = \{J: J \geq n + 1\} \quad (\text{A3.5})$$

and

$$B_1 = \{J: J \leq n\}, \quad (\text{A3.6})$$

and because there are at least  $n + 1$  zeros when  $k \leq I$ , and at most  $n$  zeros when  $k > I + 1$  in a window centered at time  $k$ .

As a second example, let us consider the Wilcoxon or the  $\text{GW}_2(N, N)$  filter. For the Wilcoxon filter  $N_e = N(N + 1)/2$ , and we find that

$$B_0 = \{J: J(J + 1) \geq N(N + 1)/2 + 1\} \quad (\text{A3.7})$$

and

$$B_1 = \{J: (N - J)(N - J + 1) \geq N(N + 1)/2 + 1\}, \quad (\text{A3.8})$$

again assuming that  $N_e$  is an odd number. Now let  $Q$  be the smallest positive integer satisfying

$$Q(Q + 1) \geq 2n^2 + 3n + 2. \quad (\text{A3.9})$$

When  $k \leq I - (Q - n) + 1$ , there are at least  $n + I - \{I - (Q - n) + 1\} + 1 = Q$  zeros in  $x^k$ , or equivalently  $J \geq Q$ . We therefore have  $y_k = 0$ . When  $k \geq I + (Q - n)$ , there are least  $n + \{I + (Q - n)\} - (I + 1) + 1 = Q$  ones in  $x^k$ , or  $N - J \geq Q$  and consequently  $y_k = 1$ . Finally when  $I - (Q - n) + 2 \leq k \leq I + (Q - n) - 1$ , we conclude that  $y_k = 1/2$  from (A3.1). Typical sets of values  $(n, N, N_e, Q)$  for the Wilcoxon filter are as follows: (2, 5, 15, 4), (4, 9, 45, 7), and (6, 13, 91, 10). Thus, we may conclude, for example, that for  $N = 9$ , the Wilcoxon filter will transform an ideal edge into a three level rise from 0 to 1 with four middle values of  $1/2$ .

#### APPENDIX D

##### AN ANALYSIS OF NARROW PULSE PERFORMANCE OF THE $\text{GW}_2(2, N)$ FILTER

###### A. For Pulses of Width $w \geq n + 1$

To show that the  $\text{GW}_2(2, N)$  filter retains pulses of width  $w \geq n + 1$ , all we have to do is to show that the  $\text{GW}_2(2, N)$  filter retains a pulse of width  $w = n + 1$ . When the window is centered at time  $k \leq I + 1$  or  $k \geq I + n + 1$ , we can conclude that  $y_k = 0$  using the result of Appendix C, the pulse looking like an ideal edge to the  $\text{GW}_2(2, N)$  filter. We then note that the possible index pairs  $(i, j)$  are  $(i, i)$  and  $(i, i + 1)$  for  $i = 1, 2, \dots, N - 1$  and  $(N, N)$ . Now when  $I + 2 \leq k \leq I + n$ ,  $x^k$  has  $n + 1$  contiguous ones and  $n$  zeros with two edges

present. Therefore, the number of ones, halves, and zeros in  $z^k$  are  $n + 1 + n = 2n + 1$ , 2, and  $2n - 2$  (since the total number of pairs is  $4n + 1$ ), respectively. From this observation, we have  $y_k = 1$ .

###### B. For Pulses of Width $w = n$

Next let the width of a pulse be  $w = n$ . Let the sequence of a pulse of width  $w = n$  be  $\{ \dots 000111 \dots 1100 \dots \}$  with the first one and the last one occurring at  $k = I + 1$  and at  $k = I + n$ , respectively. When  $k \leq I$  or when  $k \geq I + n + 1$ , we have  $y_k = 0$ , using the result of Appendix C. We then note that the possible index pairs  $(i, j)$  are  $(i, i)$  and  $(i, i + 1)$  for  $i = 1, 2, \dots, N - 1$  and  $(N, N)$ . Now when  $I + 1 \leq k \leq I + n$ ,  $x^k$  has  $n$  contiguous ones and  $(n + 1)$  zeros with two edges present. Therefore, the number of ones, halves, and zeros in  $z^k$  are  $n + n - 1 = 2n - 1$ , 2, and  $2n$  (since the total number of pairs is  $4n + 1$ ), respectively. From this observation, we have  $y_k = 1/2$ , which implies that the  $\text{GW}_2(2, N)$  filter will reduce the height of a pulse of width  $w = n$  by a factor of 2.

#### REFERENCES

- [1] L. S. Davis and A. Rosenfeld, "Noise cleaning by iterated local averaging," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-8, pp. 705-710, Sept. 1978.
- [2] D. Harwood *et al.*, "A new class of edge-preserving smoothing filters," TR-1397, Comput. Vision Lab., Univ. Maryland, pp. 1-22, Sept. 1978.
- [3] 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. 1342-1349, Dec. 1983.
- [4] Y. H. Lee and S. A. Kassam, "Generalized median filtering and related nonlinear filtering techniques," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-33, pp. 672-683, June 1985.
- [5] S. R. Peterson and S. A. Kassam, "General order statistic-based structures for nonlinear signal restoration," in *Proc. 20th Annu. Conf. Inform. Sci., Syst.*, Princeton, NJ, Mar. 1986, pp. 723-728.
- [6] P. Heinonen, J. Astola, and Y. Neuvo, "Median type filters with linear substructures," in *Proc. 1986 IEEE ISCAS*, May 1986, pp. 357-370.
- [7] I. Pitas and A. N. Venetsanopoulos, "Nonlinear mean filters in image processing," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, pp. 573-584, June 1986.
- [8] —, "Edge detectors based on nonlinear filters," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-8, pp. 538-549, July 1986.
- [9] A. Nieminen, P. Heinonen, and Y. Neuvo, "A new class of detail-preserving filters for image processing," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-9, pp. 74-90, Jan. 1987.
- [10] L. R. Rabiner, M. R. Sambur, and C. E. Schmidt, "Applications of a nonlinear smoothing algorithm to speech processing," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-23, pp. 552-557, Dec. 1975.
- [11] N. S. Jayant, "Average and median-based smoothing techniques for improving speech quality in the presence of transmission errors," *IEEE Trans. Commun.*, vol. COM-24, pp. 1043-1045, Sept. 1976.
- [12] E. Ataman, B. 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.
- [13] N. C. Gallagher 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.
- [14] K. Oflazer, "Design and implementation of single chip 1-D median filter," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-31, pp. 1164-1168, Oct. 1983.
- [15] F. R. Hampel *et al.*, *Robust Statistics*. New York: Wiley, 1986.
- [16] J. Hajek and Z. Sidak, *Theory of Rank Tests*. New York: Academic, 1967.

- [17] T. P. Hettmansperger, *Statistical Inference Based on Ranks*. New York: Wiley, 1984.
- [18] P. J. Huber, *Robust Statistics*. New York: Wiley, 1981.
- [19] R. J. Crinon, "The Wilcoxon filter: A robust filtering scheme," in *Proc. ICASSP*, Tampa, FL, Mar. 1985, pp. 18.5.1-18.5.4.
- [20] J.-H. Lee and J.-Y. Kao, "A fast algorithm for two-dimensional Wilcoxon filtering," in *Proc. ISCAS*, Philadelphia, PA, May 1987, pp. 268-271.
- [21] J. L. Hodges and E. L. Lehmann, "Estimates of location based on rank tests," *Ann. Math. Stat.*, vol. 34, pp. 598-611, 1963.
- [22] D. F. Bauer, "Constructing confidence sets using rank statistics," *J. Amer. Statist. Assoc.*, vol. 67, pp. 687-690, 1972.
- [23] E. L. Lehmann, *Theory of Point Estimation*. New York: Wiley, 1983.
- [24] E. P. Markowski and T. P. Hettmansperger, "Inference based on simple rank step score statistics for the location model," *J. Amer. Statist. Assoc.*, vol. 77, pp. 901-907, 1982.
- [25] I. Song, "Nonlinear techniques for detection and filtering of discrete-time signals," Ph.D. dissertation, Dep. Elec. Eng., Univ. Pennsylvania, Philadelphia, Jan. 1987.
- [26] B. I. Justusson, "Median filtering: Statistical properties," in *Two-Dimensional Digital Signal Processing II*, T. S. Huang, Ed. New York: Springer-Verlag, 1981.
- [27] S. R. Peterson, Y. H. Lee, and S. A. Kassam, "Some statistical properties of alpha-trimmed mean and standard type  $M$  filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, pp. 707-713, May 1988.

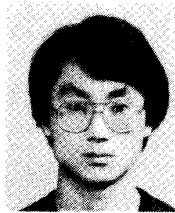


**Prashant P. Gandhi** (S'87) received the B.S.E.E. degree (magna cum laude) from the University of Maryland, College Park, in 1984, and the M.S.E.E. degree from the University of Pennsylvania, Philadelphia, in 1986.

Currently he is a candidate for the Ph.D. degree in electrical engineering at the University of Pennsylvania. He has been a Graduate Research Fellow since June 1986. His research interests are in nonlinear signal processing, detection and estimation theory, and image and radar signal processing.

Mr. Gandhi

was a recipient of a Graduate Fellowship at the University of Pennsylvania from January 1985 to May 1986.



**Tackho Song** (S'80-M'87) was born in Seoul, Korea, on February 20, 1960. He received the B.S. (magna cum laude) and the M.S.E. degrees in electronics engineering from Seoul National University, Seoul, Korea, in 1982 and 1984, respectively. He also received the M.S.E. and Ph.D. degrees in electrical engineering from the University of Pennsylvania, Philadelphia, in 1985 and 1987, respectively.

He was a Research Assistant at the University of Pennsylvania during 1984-1987, actively engaged in research in noiseless image coding and statistical techniques for signal processing. He was a member of the Technical Staff at Bell Communications Research, Morristown, NJ, in 1987. He is now an Assistant Professor at the Department of Electrical and Electronics Engineering, Korea Advanced Institute of Science and Technology (KAIST), Seoul, Korea. His research interests include detection and estimation theory, statistical signal and image processing, and statistical communication theory.

Dr. Song was awarded a University Scholarship from Seoul National University during 1978-1983. He was the Student Chairman of the IEEE Student Activities Committee, Seoul National University Branch, in the academic year 1982-1983. He was a recipient of the Korean Honor Scholarship in 1985 and 1986, and the Korean American Scholarship in 1986. He is also a member of the Acoustical Society of Korea (ASK), the Korean Institute of Communication Sciences (KICS), the Korean Institute of Telematics and Electronics (KITE), and the Korean Scientists and Engineers Association in America (KSEA).



**Saleem A. Kassam** (S'74-M'76-SM'83) was born in Dar-es-Salaam, Tanzania, on June 16, 1949. He received the B.S. degree in electrical engineering (with distinction) from Swarthmore College, Swarthmore, PA, in 1972, the M.S.E. and M.A. degrees in electrical engineering from Princeton University, Princeton, NJ, in 1974, and the Ph.D. degree in electrical engineering from Princeton in 1975.

He has been with the Moore School of Electrical Engineering, University of Pennsylvania, Philadelphia, since 1975, where he is currently Professor of Electrical Engineering. His research interests have been in the areas of signal processing and communication theory, and he has made contributions specifically in signal detection, quantization, robust signal processing, spectrum estimation, image processing, and spatial array processing. He is actively engaged in research in these areas, as well as in microwave imaging and ultrasonic imaging, and also acts as a consultant to industry. He is the Co-Editor of a book, *Nonparametric Detection: Theory and Applications* (with J. B. Thomas, 1980) and author of *Signal Detection in Non-Gaussian Noise* (New York: Springer-Verlag, 1988).

Dr. Kassam is a member of Phi Beta Kappa and Sigma Xi.