LUM Filters: A Class of Rank-Order-Based Filters for Smoothing and Sharpening

Russell C. Hardie, *Member, IEEE,* and Charles G. Boncelet, *Member, IEEE*

Abstract—**In this paper, we present a new class of rank-order-based filters, lower-upper-middle (LUM) filters. The output of these filters is determined by comparing a lower- and upper-order statistic to the middle sample in the filter window. These filters can be designed for smoothing, sharpening, and outlier rejection. This wide range of characteristics can be obtained from a single filter structure by simply varying the filter parameters. Thus, this class of filters is extraordinarily versatile. When used as smoothers, LUM filters can take on a range of smoothing characteristics. The level of smoothing done by the filter can range from no smoothing to that of the median. This flexibility allows the LUM filter to be designed to best balance the tradeoffs between noise smoothing and signal detail preservation. LUM filters can be designed to enhance edge gradients. We demonstrate that they avoid many of the shortcomings of linear edge-enhancing filters. Namely, LUM filters can be designed to be insensitive to low levels of additive noise and can be designed to remove impulsive noise while enhancing edges. Furthermore, LUM filters do not cause overshoot or undershoot. We develop some statistical and deterministic properties of the LUM filters and present a number of experimental results to illustrate the performance of these filters. These experiments include applying the new filters to 1-D signals and images.**

I. Introduction

We introduce a new class of rank-order-based filters, lower-upper-middle (LUM) filters, for use in a variety of signal and image-processing applications. The name of these filters follows since a lower- and upper-order statistic are compared with the middle sample in the filter window to determine the output. More will be said about this in Section II. With an appropriate choice of parameters, LUM filters can function as smoothers, sharpeners, and outlier rejectors. Furthermore, LUM filters can be designed to sharpen edges and reject outliers at the same time.

Rank-order-based filters are widely used to do smoothing. The first, and probably still the most widely used, is a simple median of the points within a window. Median filters and their properties have been described in [3], [12], [15], [34]. In many applications, the median introduces too much smoothing. The blurring introduced may be more objectionable than the original noise. There have been many generalizations of the median, including weighted medians [15], multistage medians [1], [4], [21], [27], stack filters [9], [35], morphological filters [25], [33], linear combinations of order statistics [5], [8], [19], [30]–[32], and hybrids [14]. A subclass of LUM filters, which we refer to as LUM smoothers, have been shown in [16], [17] to be equivalent to center-weighted medians [15] and "Winsorizing" smoothers [24].

The smoothing characteristics of the LUM filter are controlled by one of two filter parameters. Varying this parameter changes the level of smoothing from no smoothing to that of the median. Having such control allows one to best balance the tradeoffs between noise smoothing and signal-detail preservation. If this parameter is selected such that a small amount of smoothing is performed, then the filter functions as an outlier rejector. It will remove outliers while leaving most "normal" samples unmodified.

LUM filters can also be designed to enhance edge gradients. The amount of enhancement done by the LUM filter is controlled by the second filter parameter. Edge enhancement and sharpening have traditionally been accomplished using linear techniques. These techniques include Wiener filtering, high-pass filtering, and unsharp masking. There are many situations where linear sharpeners do a good job. However, there are just as many where linear sharpeners do not. For instance, linear sharpeners often cause severe overshoot and undershoot—ringing—at edges and tend to amplify background noise. We will show that LUM filters avoid many of the shortcomings of conventional linear edge-enhancing filters. In particular, LUM filters can be designed to be insensitive to low levels of additive noise and can be designed to remove impulsive-type noise while simultaneously enhancing edges. Furthermore, LUM filters do not cause any overshoot or undershoot.

Although most of the recent work in nonlinear rank-order-based filters has focused on the smoothing problem, some rank-order-based sharpeners have been proposed. The comparison and selection filter (CS filter) is one such rank-order-based sharpener [18]. Other edge-enhancing filters that incorporate ranking operations are presented in [20], [22], [23]. Rank-order sharpeners generally offer resistance to outliers. However, they differ in other characteristics. The CS filter offers good performance in certain situations but can obliterate small details and distort "thin" objects. LUM sharpeners have excellent detail-

Manuscript received January 17, 1991. Revised April 16, 1992.

R. C. Hardie was with the Department of Electrical Engineering, University of Delaware, Newark, DE 19716. He is now with Earth Satellite Corporation, Rockville, MD 20852.

C. G. Boncelet is with the Department of Electrical Engineering, University of Delaware, Newark, DE 19716.

IEEE Log Number 9206037.

preserving characteristics. They can pass small details with minimal distortion.

In summary, the LUM-filter structure has two parameters, one for smoothing and one for sharpening. For a given window size and shape, these parameters can be adjusted independently to yield a wide range of characteristics. We believe that the LUM filter is extraordinarily versatile without sacrificing simplicity. Furthermore, we believe that the LUM filter is easy to understand, making the choice of parameters simple and intuitive. Practitioners can use the LUM filter with confidence and adjust its performance for specific applications.

The material in this paper is presented as follows. In Section II, we define the LUM filters. In Section III, we develop some statistical and deterministic properties of the filters. In particular, we present formulas for breakdown probabilities and properties relating to sharpening and root-signal analysis. In Section IV, we present the results of several experiments. This section contains a number of revealing filtered images and corresponding quantitative analysis and comparisons with other methods. Finally, the paper closes with conclusions in Section V.

II. The LUM Filter

In this section we define the LUM filters. We first define a subclass that we refer to as LUM smoothers. This subclass is the same as the "Winsorizing" smoothers introduced in [24]. Furthermore, it has been shown in [16], [17] that these smoothers are equivalent to center-weighted medians. We then define a second subclass of filters, the LUM sharpeners, which can enhance edges. Finally, we define the general class of LUM filters that includes LUM smoothers and LUM sharpeners as special cases. Also included in this class are filters that can simultaneously enhance edges and suppress impulsive noise.

Consider a window function containing a set of N samples centered about the sample x^*. We assume N to be odd. This set of observations will be denoted $\boldsymbol{W} = \{x_1, x_2, \cdots, x_N\}$. For a 2-D signal (e.g., an image), let the window be a simple $(2m + 1) \times (2m + 1) = N$ square. The rank-ordered set can be written as

$$x_{(1)} \leq x_{(2)} \leq \cdots \leq x_{(N)}. \tag{1}$$

The estimate of the center sample will be denoted y^*.

A. The LUM Smoother

The LUM smoother is defined below.

Definition 1: The output of the LUM smoother with parameter k is given by

$$y^* = \text{med}\,\{x_{(k)}, x^*, x_{(N-k+1)}\} \tag{2}$$

where $1 \leq k \leq (N + 1)/2$. ■

Thus, the output of the LUM smoother is $x_{(k)}$ if $x^* < x_{(k)}$. If $x^* > x_{(N-k+1)}$, then the output of the LUM smoother is $x_{(N-k+1)}$. Otherwise the output of the LUM smoother is simply x^*.

The reasoning behind comparing the middle sample x^* to the lower- and upper-order statistics is that these order statistics form a range of "normal"-valued samples. If x^* lies in this range it is not modified. If x^* lies outside this range it is replaced by a sample that lies closer to the median. This creates a smoothing function. For example, if x^* is an impulse it is likely to fall outside the range of the upper- and lower-order statistics. It would then be replaced with a value closer to that of the median, and the outlier would be removed.

The tuning parameter k controls the smoothing characteristics of the filter and can be adjusted to best balance the tradeoffs between noise smoothing and signal-detail preservation. Note that for $k = (N + 1)/2$ the output of the LUM smoother is the median of $\boldsymbol{W}$, and this is the maximum amount of smoothing that can be performed. As k is decreased the filter exhibits improved detail-preserving characteristics; when $k = 1$, the LUM smoother becomes an identity filter (i.e., $y^* = x^*$).

As stated earlier, LUM smoothers are equivalent to center-weighted medians. Center-weighted medians are a special case of weighted medians [15] and are given by the median over a modified set of observations containing multiple center samples. Specifically, center-weighted medians are defined by

$$y^* = \text{med}\,\{\boldsymbol{W} \cup \underbrace{\{x^*, x^*, \cdots, x^*\}}_{w-1}\} \tag{3}$$

where w is the weight of the center sample and is assumed to be an odd positive integer. The relationship between the parameter w in the center-weighted median and the parameter k in the LUM smoother is

$$w = N - 2k + 2 \tag{4}$$

where $1 \leq k \leq (N + 1)/2$.

Implementing the center-weighted median using the structure in (2) leads to a different intuitive understanding of the filtering operation. Also, implementation of the center-weighted median as shown in (2) requires fewer operations than that of (3), since fewer elements must be sorted. Specifically, in Oflazor's odd-even transposition sort [29], (2) requires approximately N^2 computing elements while (3) requires $(N + w - 1)^2$.

B. The LUM Sharpener

The LUM smoother and other rank-order smoothers obtain smoothing characteristics by shifting samples toward the median. To obtain sharpening characteristics, samples must be moved away from the median to more extreme-order statistics. This is how the LUM sharpener operates.

Before defining the LUM sharpener, let us first define a value centered between the lower- and upper-order statistics, $x_{(l)}$ and $x_{(N-l+1)}$. This midpoint or average, denoted t_l, is given by

$$t_l = (x_{(l)} + x_{(N-l+1)})/2. \qquad (5)$$

We define the LUM sharpener as follows.

Definition 2: The output of the LUM sharpener with parameter l is given by

$$y^* = \begin{cases} x_{(l)}, & \text{if } x_{(l)} < x^* \leq t_l \\ x_{(N-l+1)}, & \text{if } t_l < x^* < x_{(N-l+1)} \\ x^*, & \text{otherwise.} \end{cases} \qquad (6)$$

■

Thus, if $x_{(l)} < x^* < x_{(N-l+1)}$, then x^* is shifted outward to $x_{(l)}$ or $x_{(N-l+1)}$ according to which is closest to x^*. Otherwise the sample x^* is unmodified. The reasoning behind this sharpening operation is that if $x_{(l)} < x^* < x_{(N-l+1)}$ then x^* is interpreted as being a transition sample in a slope region. By shifting it to an extreme-order statistic we are removing the transition points and creating a steeper slope and in many cases an ideal step edge.

By changing the parameter l, various levels of sharpening can be achieved. In the case where $l = (N + 1)/2$, no sharpening occurs and the LUM sharpener is simply an identity filter. In the case where $l = 1$, a maximum amount of sharpening is achieved since x^* is being shifted to one of the extreme-order statistics $x_{(1)}$ or $x_{(N)}$.

The CS filter [18] also obtains its sharpening characteristics by shifting samples away from the median. The CS filter is defined by

$$y^* = \begin{cases} x_{(j)}, & \text{if } \hat{\mu} \geq x_{((N+1)/2)} \\ x_{(N-j+1)}, & \text{otherwise} \end{cases} \qquad (7)$$

where $\hat{\mu}$ is the sample mean estimate of $\boldsymbol{W}$ and $1 \leq j \leq (N + 1)/2$. This parameter j is a tuning parameter that can be used to give the filter varying levels of enhancement. Note that if $j = (N + 1)/2$ the output of the CS filter is the median of $\boldsymbol{W}$.

One difficulty with the CS filter is that it distorts or removes small-signal features. On the other hand, we will show that the LUM sharpener has excellent detail preserving and enhancing characteristics.

C. The LUM Filter

To obtain an enhancing filter that is robust and can reject outliers, the philosophies of the LUM smoother and LUM sharpener must be combined. This leads us to the general LUM filter. Before we define this filter, let us first define a lower and upper statistic as follows

$$x^L = \text{med } \{x_{(k)}, x^*, x_{(l)}\} \qquad (8)$$

$$x^U = \text{med } \{x_{(N-k+1)}, x^*, x_{(N-l+1)}\} \qquad (9)$$

where $1 \leq k \leq l \leq (N + 1)/2$. Note that $x^L \leq x^U$. The output of the general LUM filter is given by the statistic x^U or x^L that is closest to the center sample x^*. The parameters k and l can be considered tuning parameters that allow the LUM filter to have a variety of characteristics. More will be said below about these parameters.

Two mathematically equivalent, but slightly different, definitions of the LUM filter are given below. Definition 3b is perhaps the most direct.

Definition 3a: The output of the LUM filter is given by

$$y^* = \begin{cases} x^L, & \text{if } x^* \leq (x^L + x^U)/2 \\ x^U, & \text{otherwise.} \end{cases} \qquad (10)$$

■

Definition 3b: An alternative definition of the LUM filter is

$$y^* = \begin{cases} x_{(k)}, & \text{if } x^* < x_{(k)} \\ x_{(l)}, & \text{if } x_{(l)} < x^* \leq t_l \\ x_{(N-l+1)}, & \text{if } t_l < x^* < x_{(N-l+1)} \\ x_{(N-k+1)}, & \text{if } x_{(N-k+1)} < x^* \\ x^*, & \text{otherwise} \end{cases} \qquad (11)$$

where t_l is the midpoint between $x_{(l)}$ and $x_{(N-l+1)}$ defined in (5). ■

Regardless of how it is implemented, the LUM filter performs a relatively simple function. The LUM outputs the sample $x_{(k)}$ if $x^* < x_{(k)}$. Similarly, if $x^* > x_{(N-k+1)}$ the output of the LUM filter is $x_{(N-k+1)}$. Thus, just like in the LUM smoother, the samples $x_{(k)}$ and $x_{(N-k+1)}$ form bounds on the LUM-filter output. On the other hand, if $x_{(l)} < x^* < x_{(N-l+1)}$ then the center sample is shifted outward to $x_{(l)}$ or $x_{(N-l+1)}$ according to which of these lies closest to x^*, as in the LUM sharpener. If x^* lies such that $x_{(k)} \leq x^* \leq x_{(l)}$ or $x_{(N-l+1)} \leq x^* \leq x_{(N-k+1)}$, then the sample is unmodified by the filtering operation.

By manipulating the parameters k and l, the LUM filter takes on a variety of characteristics. In order to illustrate this and clarify the LUM-filter definition, Fig. 1 depicts the filtering process in some special cases and in the general case. These figures represent a set of order statistics observed from some process. The samples range from small to large going left to right. The shaded statistics represent a range of values the center sample can have without being modified.

Fig. 1(a) shows the case where $l = (N + 1)/2$ and k is varied. Here the filter functions as the LUM smoother. As k is increased more smoothing can be expected and if $k = l = (N + 1)/2$ then the output of the LUM filter is simply the median. Figure 1(b) depicts the case where $k = 1$ and l is varied. In this case we get the LUM sharpener. As l is decreased, $x_{(N-l+1)}$ and $x_{(l)}$ moves toward the upper- and lower-extreme values, respectively. This leads to an increased enhancing effect. Figure 1(c) shows the case where $1 < k \leq l < (N + 1)/2$. In this case sharpening and outlier rejection can be achieved simultaneously. The parameter k can be increased to improve the impulse-rejection characteristics of the LUM filter.

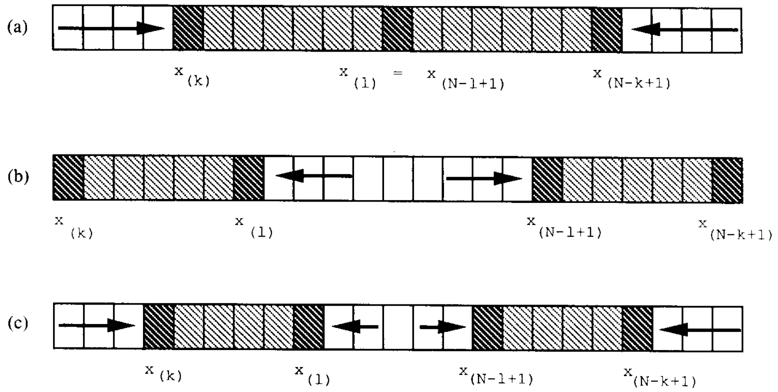


Fig. 1. LUM filter operation as the (a) LUM smoother, (b) LUM sharpener, and (c) hybrid smoothing and sharpening filter.

The parameter l, on the other hand, controls the level of enhancement. In particular, l is decreased to give more enhancement and increased to reduce enhancement.

D. The Asymmetric LUM Filter

In the definitions above, the lower- and upper-order statistics defined by the parameters k and l have been restricted to be symmetric about the median. This restriction is appropriate when the signal and noises are symmetric. However, there are applications where this is not so. For instance, the noise may contain only positive impulses. In such a case the filter need only smooth out these large-valued outliers. Also, it may be desirable to enhance only one side of edges. These types of asymmetric operations can be performed by the asymmetric LUM filter.

The asymmetric LUM filter has four parameters such that

$$x^L = \text{med}\,\{x_{(k)}, x^*, x_{(l)}\} \tag{12}$$

$$x^U = \text{med}\,\{x_{(q)}, x^*, x_{(r)}\} \tag{13}$$

and the only restriction on the parameters is that $1 \leq k \leq l \leq q \leq r \leq N$.

In the example mentioned above, where only large-valued outliers occur, the noise can be filtered with the asymmetric LUM filter by setting $k = 1$ and setting $r < N$. However, using symmetric order statistics works well in general and gives the LUM filter only two parameters. In the remainder of this paper, we will consider only the symmetric filter.

III. Properties of the LUM Filter

In this section a number of statistical and deterministic properties of the LUM filter are presented. These properties aid in the design and analysis of the filter. In the first subsection some statistical properties are presented. The second subsection develops some deterministic properties.

A. Statistical Properties

An important consideration in the design of the LUM filter is the impulsive-noise breakdown probability, introduced in [24]. Other statistical properties of order statistics and related filters can be found in [7], [10], [11]. The breakdown probability is the probability of outputting an impulse given a certain probability of impulses appearing in the input. Given these breakdown probabilities, the parameters k and l in the LUM filter can be selected to achieve a desired level of impulse rejection. Consider the case of impulsive noise having a value of either $+\infty$ or $-\infty$. Let the probability of a positive impulse and the probability of a negative impulse be $p/2$. Thus, the overall probability of an impulse occurring is p. In this case the breakdown probabilities for the LUM smoother, the LUM sharpener, and the general LUM filter are given in Properties 1, 2, and 3, respectively.

Property 1 (Breakdown Probability for the LUM Smoother): The breakdown probability for the LUM smoother is given by

$$p_b = p \sum_{i=k-1}^{N-1} \binom{N-1}{i} \left(\frac{p}{2}\right)^i \left(1 - \frac{p}{2}\right)^{N-i-1} + (2-p) \sum_{i=N-k+1}^{N-1} \binom{N-1}{i} \left(\frac{p}{2}\right)^i \cdot \left(1 - \frac{p}{2}\right)^{N-i-1}. \tag{14}$$

(For notational convenience it is assumed that if the upper index of a summation is less than the lower index, then that summation equals zero.)

Proof: The probability of the LUM smoother outputting an impulse is given by

$$\Pr(y^* = \pm\infty)$$
$$= 2\Pr(y^* = -\infty) \tag{15}$$
$$= 2\Pr(x^* = -\infty)\Pr(x_{(k)} = -\infty | x^* = -\infty) + 2\Pr(x^* \neq -\infty)\Pr(x_{(N-k+1)} = -\infty | x^* \neq -\infty) \tag{16}$$
$$= 2\left(\frac{p}{2}\right)\Pr(\text{At least } k - 1 \text{ out of } N - 1 \text{ samples equal } -\infty) + 2\left(1 - \frac{p}{2}\right)\Pr(\text{At least } N - k + 1 \text{ out of } N - 1 \text{ samples equal } -\infty). \tag{17}$$

Summing these binomial probabilities yields (14). ■

The probabilities given by Property 1 are evaluated in the case where $N = 25$ (i.e., 5 × 5 window) and are shown in Fig. 2. From Fig. 2 it can be seen that to decrease the breakdown probabilities for the LUM smoother, the parameter k should be increased. Note that by increasing k the breakdown probabilities drop rapidly. Therefore, when p is small, a low breakdown probability can be obtained with a relatively small k. Using a low value of k may be desirable since the filter will preserve signal detail well.

Property 2 (Breakdown Probability for the LUM Sharpener): The breakdown probability for the LUM sharpener is given by

$$p_b = p + (1 - p)\sum_{i=l}^{N-l-1}\sum_{j=l}^{N-i-1} \cdot \frac{(N-1)!}{(i)!(j)!(N-i-j-1)!}(p/2)^{i+j} \cdot (1-p)^{N-i-j-l}. \tag{18}$$

Proof: The probability of the LUM sharpener outputting an impulse is given by

$$\Pr(y^* = \pm\infty)$$
$$= \Pr(x^* = \pm\infty) + \Pr(x^* \neq \pm\infty)\Pr(x_{(l)} = -\infty, x_{(N-l+1)} = \infty | x^* \neq \pm\infty) \tag{19}$$
$$= p + (1-p)\Pr(\text{At least } l \text{ samples have value } -\infty \text{ and } l \text{ have value } \infty). \tag{20}$$

The second term in (20) can be expressed as a double sum of multinomial probabilities giving rise to (18). ■

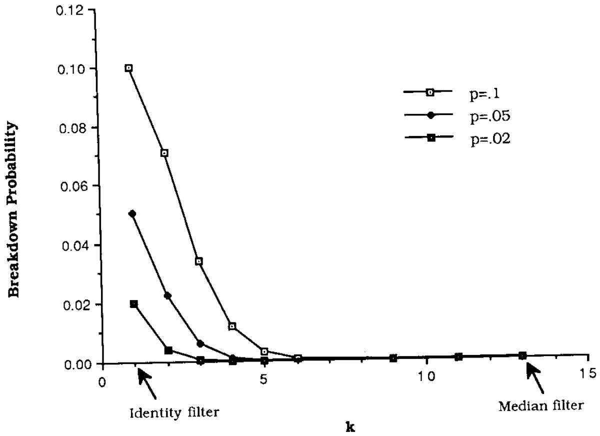


Fig. 2. Breakdown probabilities for the LUM smoother where $N = 25$.

Property 3 (Breakdown Probability for the General LUM Filter): The breakdown probability for the general LUM filter is given by

$$p_b = p\sum_{i=k-1}^{N-1}\binom{N-1}{i}\left(\frac{p}{2}\right)^i\left(1-\frac{p}{2}\right)^{N-i-1} + (2-p)\sum_{i=N-k+1}^{N-1}\binom{n-1}{i}\left(\frac{p}{2}\right)^i\left(1-\frac{p}{2}\right)^{N-i-1} + (1-p)\sum_{i=l}^{N-l-1}\sum_{j=l}^{N-i-1}\frac{(N-1)!}{(i)!(j)!(N-i-j-1)!} \cdot (p/2)^{i+j}(1-p)^{N-i-j-1}. \tag{21}$$

Proof: The probability of the LUM filter outputting an impulse is given by

$$\Pr(y^* = \pm\infty)$$
$$= 2\Pr(x^* = -\infty) \cdot \Pr(x_{(k)} = -\infty | x^* = -\infty) + 2\Pr(x^* \neq -\infty)\Pr(x_{(N-k+1)} = -\infty | x^* \neq -\infty) + \Pr(x^* \neq \pm\infty)\Pr(x_{(l)} = -\infty, x_{(N-l+1)} = \infty | x^* \neq \pm\infty). \tag{22}$$

Following directly from the proofs of Properties 1 and 2 we obtain (21). ■

From (21), it follows that impulsive noise can be combated with the general LUM filter in the same way as in the LUM smoother. Increasing the parameter k lowers the breakdown probability.

Another important property of the LUM filter is the probability that it outputs the middle sample in the filter window, in other words, the probability that a sample is unmodified by the filter operator. This probability gives some indication of the detail-preserving characteristics of the filter.

Property 4 (Probability of Outputting the Middle Sample): When the samples are independent and identically distributed (i.i.d.) with a continuous distribution, the probability that the output of the LUM filter is the middle sample in the filter window is given by

$$p_{\text{middle}} = \begin{cases} \dfrac{2(l-k+1)}{N}, & \text{for } 1 \le l < (N+1)/2 \\ \dfrac{N-2k+2}{N}, & \text{for } l = (N+1)/2. \end{cases} \tag{23}$$

Proof: The center sample x^* is unmodified by the LUM filter if $x_{(k)} \le x^* \le x_{(l)}$ or $x_{(N-l+1)} \le x^* \le x_{(N-k+1)}$. Thus, for $1 \le l < (N+1)/2$, x^* can take on $2(l-k+1)$ ranks out of N and not be modified. Given that the samples are i.i.d. and from a continuous distribution, the probability of this event is given by the first case in (23). If $l = (N+1)/2$, then x^* can take on $N - 2k + 2$ ranks out of N yielding the second case in (23). ■

From (23) it follows that as k decreases the probability of outputting the middle sample increases, giving the filter greater detail-preserving characteristics. However, this trend is at the expense of robustness, as illustrated in Fig. 2. Therefore, the value of k must be selected to balance the need for impulse rejection and detail preservation.

Note that in practice the samples may not be i.i.d. and the probability of outputting the middle sample is generally higher than that given by (23). For example, many images contain a majority of constant-valued regions. In these regions the LUM will output the middle sample. Also, in slope regions the LUM smoother will output the middle sample.

B. Deterministic Properties

In this subsection, we derive some deterministic properties of the LUM filter. First it is shown that the LUM filter has the property of scale and bias invariance and that no overshoot or undershoot can occur. Next, some classes of root signals are defined. Finally, some properties relating to sharpening are presented.

Property 5 (Scale and Bias Invariance): If the input vector $\boldsymbol{X}$ is transformed to $a\boldsymbol{X} + b\boldsymbol{1}$, then the LUM output is transformed from $\boldsymbol{Y}$ to $a\boldsymbol{Y} + b\boldsymbol{1}$ where $\boldsymbol{1}$ is a vector of 1's.

Proof: If $a > 0$ then the sorted samples in $a\boldsymbol{X} + b\boldsymbol{1}$ appear in the same order as in $\boldsymbol{X}$ regardless of the value of the offset b. In particular, the center sample bears the same relationship to the other order statistics. The results of each comparison is unchanged. Thus, the output is simply transformed to $a\boldsymbol{Y} + b\boldsymbol{1}$. If $a < 0$, then the ordering is reversed. The comparisons are still the same, except for the reversal of roles of k and $N - k + 1$, and l and $N - l + 1$. Thus, the filter's output is still transformed to $a\boldsymbol{Y} + b\boldsymbol{1}$. ■

Property 6 (No Undershoot or Overshoot): Since the output of the LUM filter is restricted to be a sample in the input set, it will never cause any undershoot or overshoot. ■

Root-signal analysis has proven to be a useful tool for evaluating nonlinear filters [2], [12], [28]. A root signal is one that is unchanged by the filtering operation. By investigating root signals one can gain insight into the performance of a given filtering algorithm. Root signals can also aid in filter design. For these reasons, we define some important classes of 1-D root signals of the LUM filter in the following two properties. We will define a 1-D sequence $x(n)$ where n can be considered a time index. Let $n \in \mathbf{Z}$ where $\mathbf{Z}$ is the set of all integers and let the filter window be centered about the sample $x(n^*)$.

Property 7 (Piecewise-Constant Root Signals): A piecewise-constant signal where each constant region is of length greater than or equal to $\max\{k, N - 2l + 1\}$ is a root signal of the LUM filter.

Proof: Given a piecewise-constant signal with constant regions of length k or greater, the center sample $x(n^*)$ can never be *strictly* greater than $x_{(N-k+1)}$ or *strictly* less than $x_{(k)}$. This follows since at any filter-window location, at least k samples will have the value of $x(n^*)$. If the constant regions are at least $N - 2l + 1$ and $N - 2l + 1 < (N+1)/2$, then at least $N - 2l + 1$ samples will be equal to $x(n^*)$ for any n^*. Thus, the case where $x_{(l)} < x(n^*) < x_{(N-l+1)}$ will not occur since the number of samples *strictly* between $x_{(l)}$ and $x_{(N-l+1)}$ is $N - 2l$. If the constant regions are at least $N - 2l + 1$ and $N - 2l + 1 \ge (N+1)/2$, then for any n^* only two values can be contained in the filter window. For any two-valued signal, $x(n^*)$ cannot lie in the range $x_{(l)} < x(n^*) < x_{(N-l+1)}$, and consequently no sharpening will occur. Thus, if a piecewise-constant signal has constant regions greater than or equal to $\max\{k, N - 2l + 1\}$, then $x(n^*)$ will never be greater than $x_{(N-k+1)}$, less than $x_{(k)}$, or such that $x_{(l)} < x(n^*) < x_{(N-l+1)}$. Therefore, by Definition 3b, $y(n^*) = x(n^*)$ for all n^* making such a signal a root of the LUM filter. ■

Thus, from Property 7 it follows that ideal step edges will not be modified by the LUM filter if they are edges between constant-valued regions of sufficient length. Another class of root signals is defined in the following property.

Property 8 (Pulse Root Signals): A pulse of duration greater than or equal to k in a constant region is a root signal of the LUM filter.

Proof: Consider the case of a pulse preceded and followed by constant regions of value c_1. Let these constant regions extend such that, in the area of interest, the window only spans the pulse or the constant background (i.e., no other structures are spanned by the window). Let the pulse start at n_0 and have a duration of P samples with height c_2, where $c_2 > c_1$. For $n^* < n_0$ we have $x(n^*) = x_{(k)} = c_1$ since this is a constant region. Thus, by Definition 3b, $y(n^*) = x(n^*) = c_1$ in this region. When $n_0 \leq n^* \leq n_0 + P - 1$ we have $x(n^*) = c_2$, and if $P \geq k$, then $x_{(N-k+1)} = c_2$. Therefore, $x(n^*) = x_{(N-k+1)}$ and thus, by Definition 3b, $y(n^*) = x(n^*) = c_2$ in this region. For $n^* > n_0 + P - 1$ we have $x(n^*) = x_{(k)} = c_1$ since this is a constant region. Again, by Definition 3b, $y(n^*) = x(n^*) = c_1$ in this region. The case where $c_2 < c_1$ follows from Property 5 and the arguments above. ■

From Property 8 it is clear that the LUM can be designed to preserve fine detail such as a small pulse by using a low value of k, even with a large window size. On the other hand, the CS filter will remove any pulse with duration less than half the window size (i.e., less than $(N + 1)/2$) regardless of the parameter j. To show this, consider the case of a constant region of value c_1 and a pulse of length $(N - 1)/2$ with a height of c_2. In any filter position the maximum number of samples with value c_2 is $(N - 1)/2$; thus the median will always be c_1. If $c_2 > c_1$, the mean will be greater than c_1 in any filter position containing at least one pulse sample. Since the mean is greater than median, the output of the CS filter will be the lower-order statistic $x_{(j)}$, and in all filter positions $x_{(j)} = c_1$. Since the CS filter will always output c_1, the pulse will be eliminated. The case of $c_2 < c_1$ can be shown similarly.

The following properties relate to the 1-D sharpening characteristics of the LUM filter. Although 1-D signals are considered, some of the 2-D filtering characteristics can be inferred from these results. Property 9 considers the shifting of concave and convex sequences. A sequence is strictly convex if $x(n) - x(n - 1) < x(n + 1) - x(n)$ for all n. Similarly we define a strictly concave sequence as one where $x(n) - x(n - 1) > x(n + 1) - x(n)$. An ideal ramp sequence is formed when $x(n) - x(n - 1) = x(n + 1) - x(n)$.

Property 9 (Shifting Concave and Convex Sequences): A strictly increasing convex sequence or a strictly decreasing concave sequence will be shifted by the LUM filter such that

$$y(n^*) = x(n^* - (N + 1)/2 + l). \tag{24}$$

A strictly increasing concave sequence or a strictly decreasing convex sequence will be shifted such that

$$y(n^*) = x(n^* + (N + 1)/2 - l). \tag{25}$$

Proof: Assume that all of the samples within the span of the filter window comprise a strictly concave or convex sequence. In these cases $x_{(l)} < x(n^*) < x_{(N-l+1)}$. In convex regions $x(n^*) < t_l = (x_{(l)} + x_{(N-l+1)})/2$ and thus, from Definition 3b, $y(n^*) = x_{(l)}$. In an increasing convex region $x_{(l)} = x(n^* -(N + 1)/2 + l)$ and thus

$$y(n^*) = x(n^* - (N + 1)/2 + l). \tag{26}$$

Likewise, in a decreasing convex region $x_{(l)} = x(n^* + (N + 1)/2 - l)$ and thus

$$y(n^*) = x(n^* + (N + 1)/2 - l). \tag{27}$$

The concave case can be proven using the same method. ■

Often edges will be composed of convex regions connected to concave regions. According to Property 9 we see that increasing convex sequences are delayed while increasing concave ones are advanced. Thus, an edge composed of an increasing convex region followed by an increasing concave region will be "compacted" by the LUM filter creating a shorter transition region and a sharper edge. Note that if we let $l = (N + 1)/2$ so that no sharpening is performed, this edge shifting will not occur.

Property 10 (Slope Enhancement): An increasing or decreasing region of length R samples, preceded and followed by constant regions with length greater than or equal to $(N - 1)/2$ samples, will be converted to an ideal step edge in a single pass of the LUM filter if $(N - 2l + 5)/2 \geq R$. The edge is positioned at the location where the samples contained in the slope first pass the midpoint between the top and bottom of the slope.

Proof: Assume we have an increasing region with R samples starting at n_0 with a value of c_1. Let this slope be preceded by a constant region of at least $(N - 1)/2$ samples of value c_1. The slope ends at $n_0 + R - 1$ with a value of c_2. Assume it is followed by a constant region of at least $(N - 1)/2$ samples of value c_2. For $n^* = n_0$ we have $x(n^*) = x_{(l)} = c_1$ and thus, by Definition 3b, $y(n^*) = x(n^*) = c_1$. For $n^* = n_0 + R - 1$ we have $x(n^*) = x_{(N-l+1)} = c_2$ and thus, by Definition 3b, $y(n^*) = x(n^*) = c_2$. In the increasing slope region where $n_0 < n^* < n_0 + R - 1$ we have

$$x_{(l)} \leq x(n^*) \leq x_{(N-l+1)}. \tag{28}$$

Thus, the output of the LUM will be either $x_{(l)}$ or $x_{(N-l+1)}$. If

$$(N - l + 1) - l + 1 \geq 2R - 3 \tag{29}$$

then $x_{(N-l+1)}$ and $x_{(l)}$ will "straddle" the slope region such that $x_{(l)} = c_1$ and $x_{(N-l+1)} = c_2$ for $n_0 < n^* < n_0 + R - 1$. Equation (29) reduces to $(N - 2l + 5)/2 \geq R$. If this is true, then the output of the LUM filter for $n_0 < n^* < n_0 + R - 1$ is given by

$$y(n^*) = \begin{cases} c_1, & \text{if } x(n^*) \leq (c_1 + c_2)/2 \\ c_2, & \text{otherwise.} \end{cases} \tag{30}$$

The case of a decreasing signal can be proven in the same fashion. ■

The LUM filter positions a step edge at a good (and arguably optimal) location. For example, if the slope is an ideal ramp of length R, then the step edge is created at the midpoint of the ramp.

Repeated applications of the LUM filter will tend to change all slopes in a signal to step edges. These step edges are root signals of the LUM and will not be further modified by the filter operator. This is in contrast with conventional linear-enhancing filters that do not converge to any root signal. Repeated applications of linear gradient-enhancing filters often result in unstable systems where the signal may be lost or heavily distorted.

It should be noted from Property 10 that in order to convert a larger slope (large R) to a step edge in a single pass of the LUM filter, a larger window size N is required along with a small value of l. However, from Property 7 we know that a larger window may not have small step edges as root signals. Thus, some loss of small step edges may occur in an attempt to fully enhance large slopes to ideal step edges. Thus, a window size must be chosen to balance the slope enhancing power of the LUM filter with its detail-preserving characteristics.

C. Two-Dimensional Window Structures

It is interesting to note that in a 2-D signal, a variety of window structures (other than a square window) can be used to obtain different enhancement characteristics. Some examples of useful window structures are shown in Fig. 3. Fig. 3(a) shows a conventional square window. Fig. 3(b) shows a rectangular window. Such a window will cause different amounts of enhancement in the horizontal and vertical directions. Namely, by extending the result in Property 10, more enhancement can be expected in the dimension in which the filter window is largest.

Fig. 3(c) shows a 1-D window that can be applied to a 2-D signal. Using such a filter window will result in a predominantly one-directional enhancement along the axis of the filter window.

A window with a cross pattern as shown in Fig. 3(d) will have both vertical and horizontal edge enhancing power with far fewer samples than a complete square window. The enhancing power of such a window structure will be comparable to its square window counterpart if the length and height of the filter windows are the same. In particular, the number of samples in the cross window structure is $4m + 1$ as compared to $(2m + 1)^2$ for the square window. With fewer samples, the LUM algorithm will require less sorting and should therefore be significantly faster. Thus, the cross window structure is a practical alternative to a square window in cases where little or no noise is present.

When significant levels of noise are present, the square or rectangular window structure can be expected to perform better than the cross window structure. This is because some redundant samples may be present. Consider the case where a 2-D signal is a ramp in one dimension and a constant in the other. In this case, the function of square window of size $(2m + 1) \times (2m + 1)$ reduces to that of a 1-D window of length $2m + 1$. This follows since in the square window there will be $2m + 1$ identical sets of $2m + 1$ samples. Thus, $2m$ of these sets are redundant. In a noisy environment the redundant samples contained in a square window are valuable. They allow for the use of higher values of k and l (for greater insensitivity to outliers) with the same amount of enhancement.

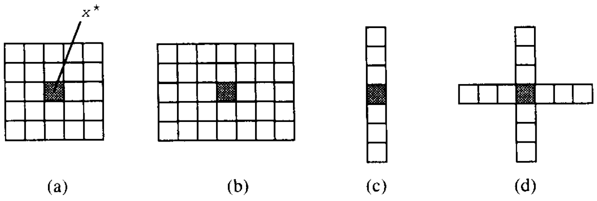


Fig. 3. Two-dimensional window structures. (a) Square. (b) Rectangular. (c) One dimensional. (d) Cross.

IV. Experimental Results

We performed a number of experiments in order to evaluate the LUM filter. The results of these experiments are given in this section. First, we present some results using 1-D signals. Next, we demonstrate the image-smoothing and impulse-rejection characteristics of the LUM filter using a standard image. Finally, we illustrate the 2-D sharpening characteristics of the LUM on a blurred test image with no noise and then with Gaussian noise.

A. One-Dimensional Signals

In this subsection we consider the filtering of a simple 1-D signal. This signal, shown in Fig. 4(a), consists of a small pulse of length 3, a medium-sized pulse that has been blurred (using a size 5 mean filter) and a large blurred pulse.

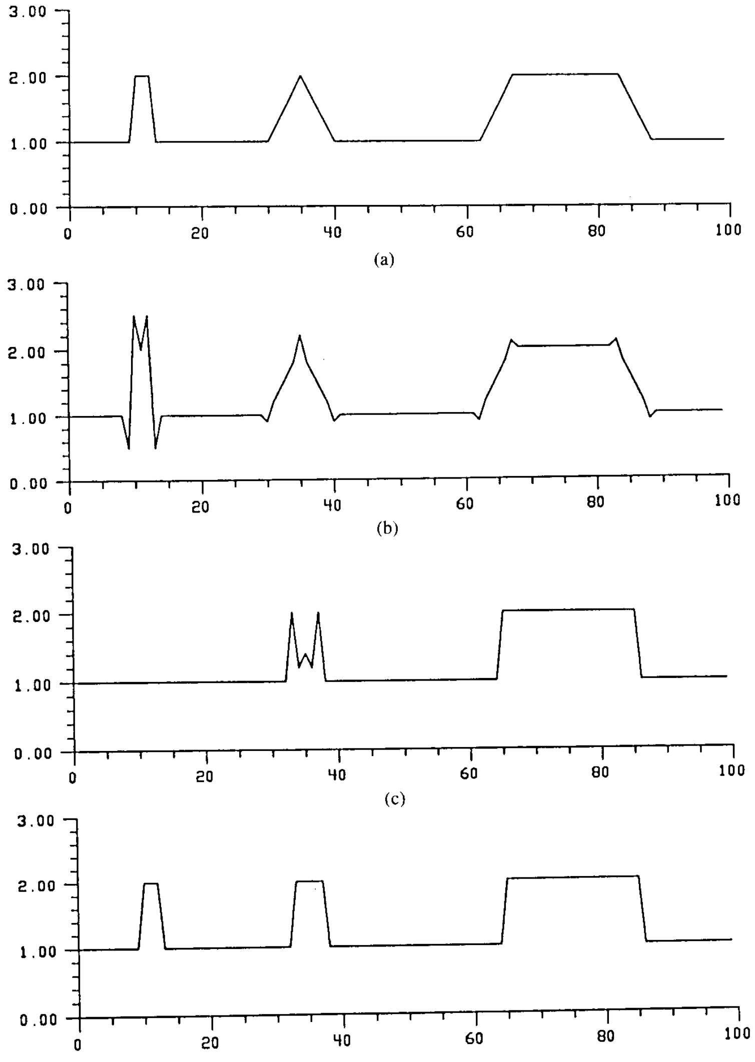
For comparison with the LUM filter we consider the conventional linear gradient-enhancing technique known as unsharp masking. The output of this linear operator is given by

$$y(n) = x(n) + s\nabla \tag{31}$$

where s is a scaling constant that determines the extent of the enhancement and ∇ is a gradient estimate. The most commonly used gradient estimate in this application is a discrete Laplacian gradient given by

$$\nabla = x(n) - \tfrac{1}{2}\{x(n - 1) + x(n + 1)\}. \tag{32}$$

Fig. 4(b) shows the signal of Fig. 4(a) filtered using unsharp masking where $s = 1$. The overshoots and undershoots caused by this technique are highly evident and the blurred edges are not fully restored to step edges. Fig. 4(c) shows the output of the CS filter with parameters $N = 7$ and $j = 1$ (maximum enhancement). Here the CS filter has removed the small pulse and distorted the medium-sized pulse. The large pulse has been fully restored to an ideal step edge. One way of preserving small struc-



(a)

(b)

(c)

(d)

Fig. 4. One-dimensional signal enhancement. (a) Input signal. (b) Output using unsharp masking where $s = 1$. (c) Output of the CS filter $N = 7, j = 1$. (d) LUM filter $N = 7$, $k = 1$, and $l = 1$.

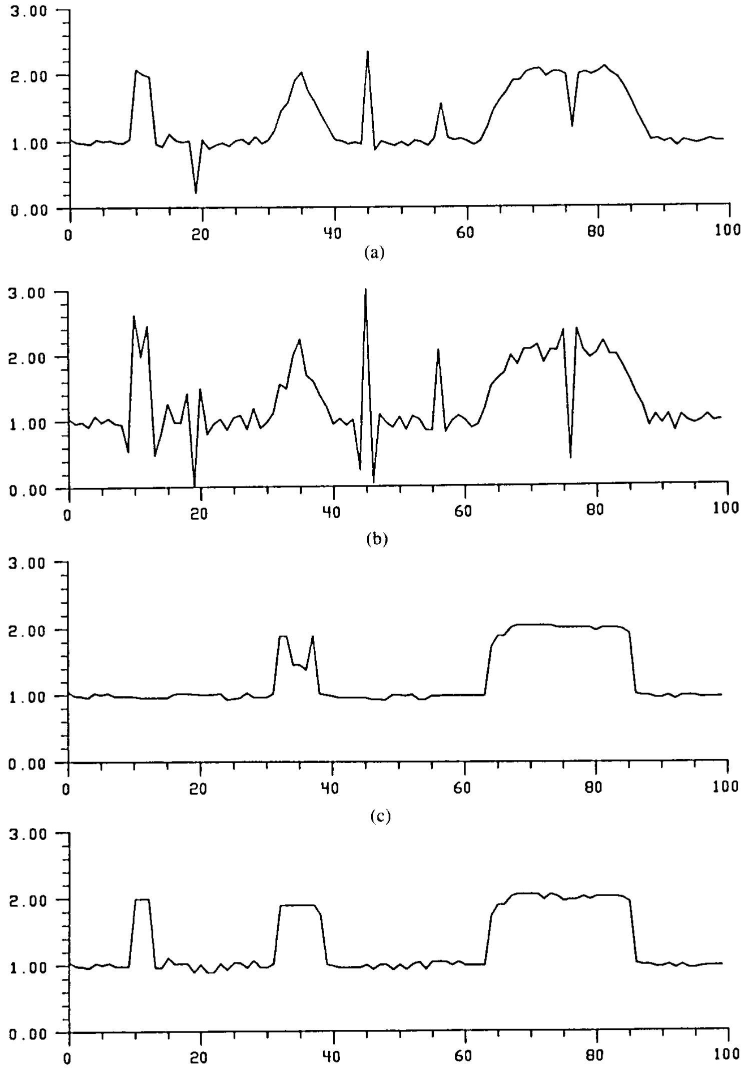
tures with the CS filter is to use a smaller filter window. However, in order to get the same level of enhancement from a smaller window multiple filter passes are required [18]. This will increase the time and computational burden of the filtering algorithm. Finally, Fig. 4(d) shows the output of the LUM filter with parameters $N = 7$, $k = 1$, and $l = 1$ (maximum enhancement). The LUM filter passed the small pulse and it fully restored the other blurred pulses to ideal steps. Also note that the LUM filter did not cause any overshoot or undershoot while enhancing the blurred pulses.

Next, we consider these filters operating on a noisy signal. We will use the contaminated Gaussian-noise model that has a probability density function given by

$$f(z) = (1 - \epsilon) \frac{1}{\sqrt{(2\pi\sigma_1^2)}} \exp\left(-\frac{z^2}{2\sigma_1^2}\right) + \epsilon \frac{1}{\sqrt{(2\pi\sigma_2^2)}} \exp\left(-\frac{z^2}{2\sigma_2^2}\right) \tag{33}$$

where $\sigma_2 > \sigma_1$. Thus, σ_1 determines the background-noise variance and σ_2 determines the variance of the impulsive type "contamination." The parameter ϵ determines the probability of an impulse.

Fig. 5(a) shows the signal of Fig. 4(a) corrupted with contaminated Gaussian noise having parameters σ_1 =



(a)

(b)

(c)

(d)

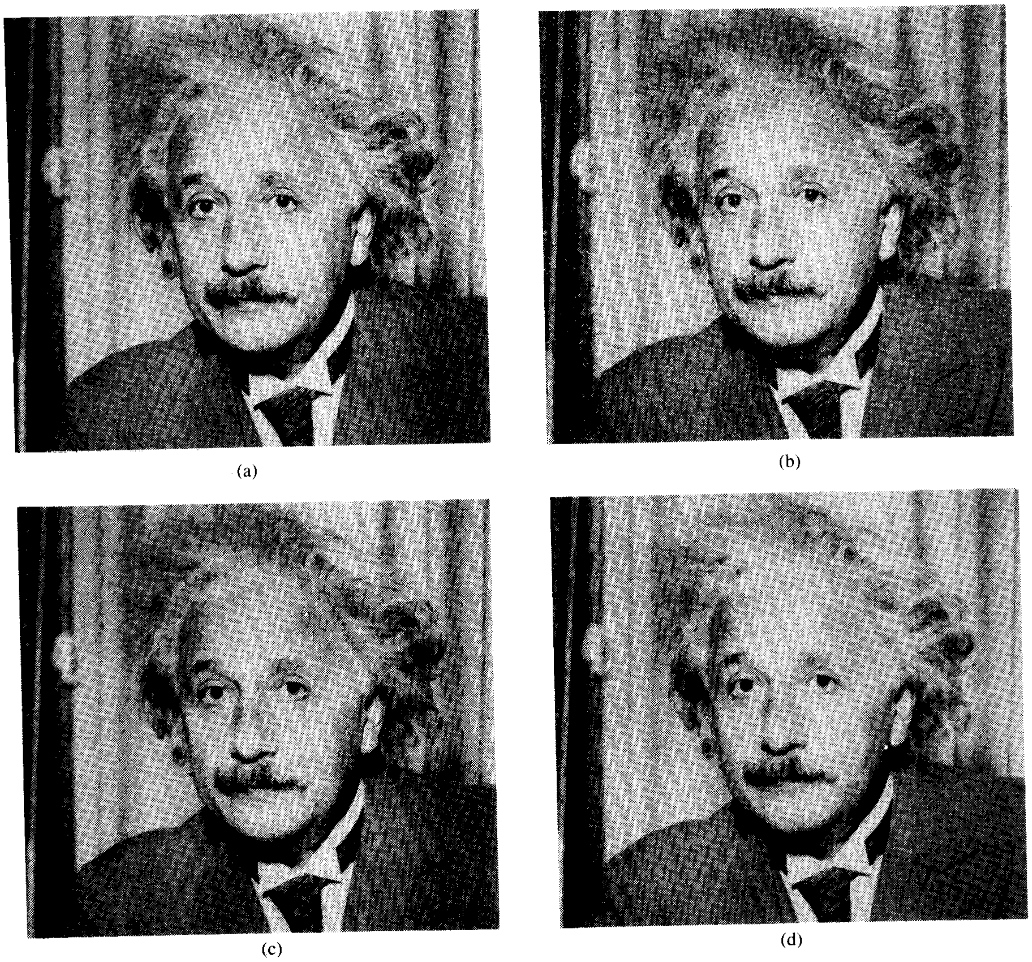
Fig. 5. Noisy one-dimensional signal enhancement. (a) Noisy input signal. (b) Output using unsharp masking where $s = 1$. (c) Output of the CS filter $N = 7$ and $j = 2$. (d) LUM filter $N = 7$, $k = 2$ and $l = 2$.

0.05, $\sigma_2 = 1$, and $\epsilon = 0.05$. Fig. 5(b) shows the signal in Fig. 5(a) filtered using unsharp masking with parameter $s = 1$. The background noise and the impulses are greatly accentuated by the unsharp masking technique. Fig. 5(c) shows the output of the CS filter with parameters $N = 7$ and $j = 2$. Again, the CS filter removes the small pulse and distorts the medium-sized pulse. However, the CS filter does a good job on the larger pulse and appears to be relatively insensitive to the additive noise. Finally, Fig. 5(d) shows the output of the LUM filter with parameters $N = 7$, $k = 2$, and $l = 2$. Again, the LUM filter preserves the small pulse and enhances the blurred pulses. Note that by using the parameters $k = 2$ and $l = 2$, the LUM filter suppresses impulses and is also relatively insensitive to the background noise.

B. *Image Smoothing and Impulse Rejection*

In this subsection we illustrate the image-smoothing and impulse-rejection characteristics of the LUM filter. Also, a quantitative comparison between the LUM filter and other techniques is presented.

Fig. 6 shows the filtering of an image corrupted with impulsive noise. Fig. 6(a) shows the original test image that has a resolution of 512 × 512 pixels with 8 bits/pixel grey-scale quantization. Fig. 6(b) shows the image cor-



(a) (b) (c) (d)

Fig. 6. Images showing the smoothing and impulse-rejection characteristics of the LUM filter. (a) Original. (b) Impulsive noise $p = 0.02$. (c) Output of the 5×5 LUM smoother where $k = 5$ and $l = 13$. (d) Output of the 3×3 median.

rupted with impulsive noise having probability $p = 0.02$. Fig. 6(c) shows the output of the LUM smoother with a 5×5 window and parameters $k = 5$ and $l = 13$. Here all of the impulses have been removed and the image detail has been preserved well. Fig. 6(d) shows the corrupted image filtered with a 3×3 median. This filtering operation has caused more smoothing than that of the LUM.

Tables I and II give a quantitative comparison between the LUM smoother and other techniques. These tables show the error for the test image shown in Fig. 6(a) with various impulsive-noise probabilities. The parameter k in the LUM smoother has been selected by using Property 1 (Fig. 2) such that k is set to be as small as possible provided that the resulting breakdown probability is less than $1/(512 \times 512)$. Thus, on average, we can expect there to be less than one impulse in the entire output image. This design criteria is visually a good one, but it is not necessarily optimal in a mean absolute or root-mean-error sense.

For comparison, we have included results using a 3×3 median (MED), a 3×3 separable median (S. MED) [26], [28], and a size 5 multistage median filter (MMF) [1] in these tables. Table I shows that the 3×3 median and 3×3 separable median actually increase the mean absolute error (MAE) of the unfiltered noisy image for $p = 0.01$ and $p = 0.02$ because of the blurring caused by these filters. On the other hand, since the impulses are removed we see a significant decrease in root-mean-squared (rms) error as shown in Table II for the same images. Note that the MMF has lower MAE than the LUM smoother for $p = 0.05$ and $p = 0.1$. This is because the MMF is doing less smoothing than the LUM with the given parameters and consequently lets impulses pass.

TABLE I
MEAN ABSOLUTE ERROR FOR TEST IMAGE WITH IMPULSIVE NOISE

	Unfiltered	5 × 5 LUM	3 × 3 MED	3 × 3 S. MED	Size 5 MMF
$p = 0.01$	1.27	(k = 4) 0.73	4.16	4.13	1.26
$p = 0.02$	2.58	(k = 5) 1.10	4.32	4.16	1.33
$p = 0.05$	6.33	(k = 7) 1.94	4.42	4.28	1.65
$p = 0.10$	12.67	(k = 9) 2.97	4.59	4.53	2.64

TABLE II
ROOT-MEAN-SQUARED ERROR FOR TEST IMAGE WITH IMPULSIVE NOISE

	Unfiltered	5 × 5 LUM	3 × 3 MED	3 × 3 S. MED	Size 5 MMF
$p = 0.01$	13.23	(k = 4) 2.84	6.54	6.41	3.41
$p = 0.02$	18.84	(k = 5) 3.53	6.58	6.47	3.72
$p = 0.05$	29.43	(k = 7) 4.79	6.71	6.66	5.86
$p = 0.10$	41.65	(k = 9) 6.30	7.10	7.40	11.36

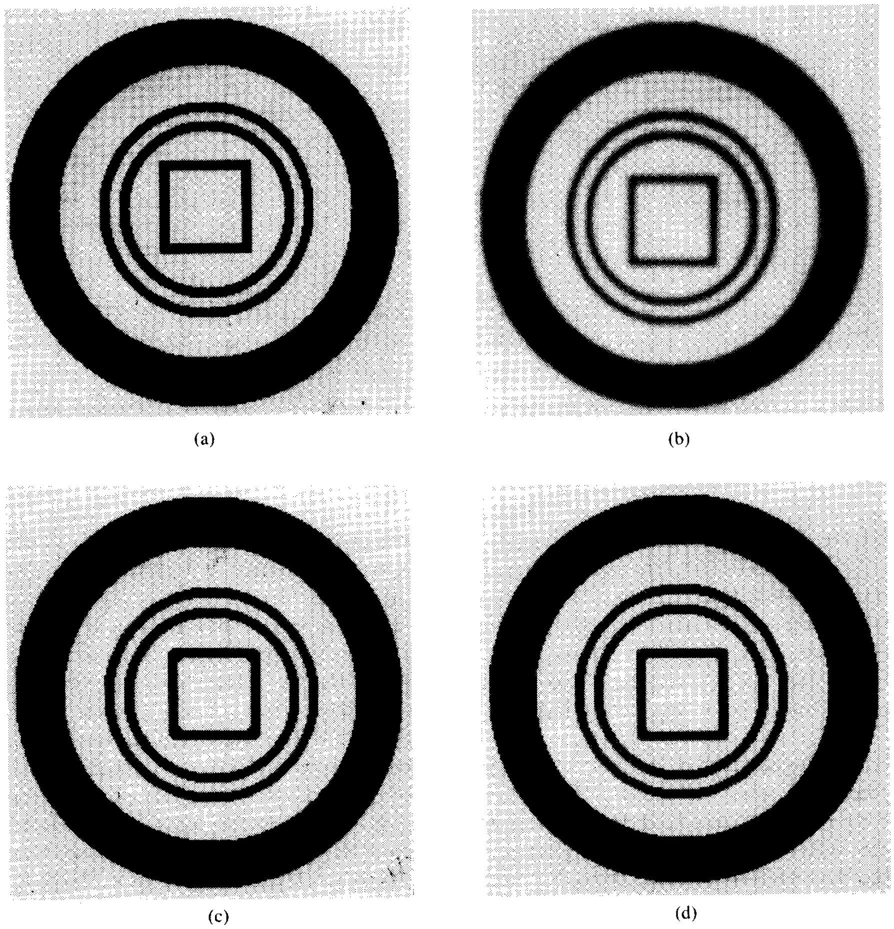


Fig. 7. Images showing the restoration of a blurred image. (a) Original test image. (b) Blurred image. (c) Output of the LUM with a 7 × 7 square window where $k = 1$ and $l = 1$. (d) Output of the LUM with a cross-window function of length 7 and height 7 where $k = 1$ and $l = 1$.

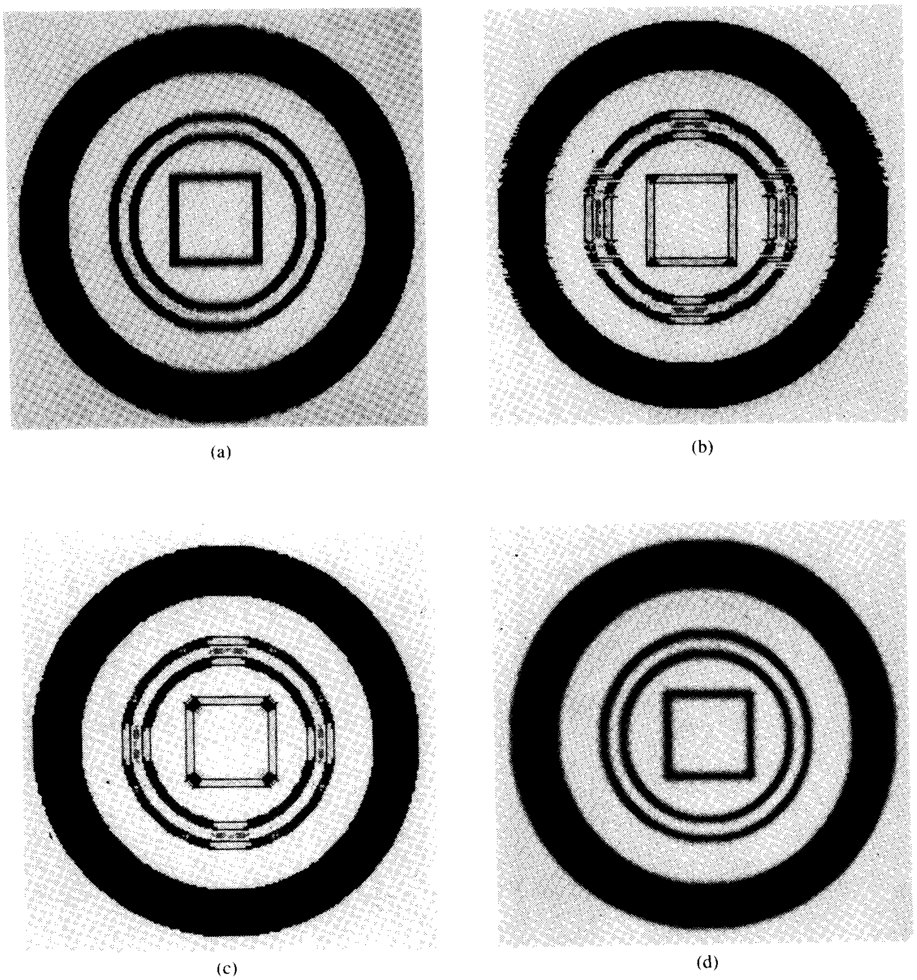


Fig. 8. Output of the (a) LUM filter with a one-dimensional horizontal window $N = 7$ and $k = l = 1$. (b) Separable CS filter where $N = 7$ and $j = 1$. (c) CS filter with a 7×7 square window $j = 1$. (d) Unsharp masking $s = 3$.

This is apparent when examining the rms errors for those images. The rms error for the MMF is much higher on these images than it is for the LUM smoother. It is interesting to note that the MAE can be substantially reduced by doing less smoothing and allowing some impulses through. For example, consider the case where $p = 0.02$. From Property 1, the LUM smoother must have $k \geq 5$ to yield an average of less than one impulse in the 512×512 output image. Using this parameter, the LUM filter has a MAE of 1.10, however, by letting $k = 3$ some impulses will get through but the MAE $= 0.69$. This trend is also true for higher impulse probabilities.

C. Image Sharpening

In this subsection, the 2-D edge-enhancing capabilities of the LUM filter are demonstrated. First we consider the LUM filter enhancing a blurred test image. Next we apply the LUM filter to a blurred test image with Gaussian noise.

Fig. 7 shows the enhancement of a blurred test image. Fig. 7(a) shows the original test image. This test image has a resolution of 256×256 with 8 bits/pixel grey-scale quantization. Fig. 7(b) shows the blurred image. The blurring function is a 5×5 mean filter. Fig. 7(c) shows the output of the LUM filter with a 7×7 square window and parameters $k = 1$ and $l = 1$ (maximum enhancement). Notice that the blurred edges have been fully restored to ideal step edges. Some distortion can be seen in the corners of the box in the center of the image. Finally, Fig. 7(d) shows the image filtered with the LUM having a cross-window function of height 7 and length 7 where $k = 1$ and $l = 1$ (maximum enhancement). This window function did an excellent job. The corners of the box have been enhanced better than those using the square window.

Fig. 8 shows other techniques enhancing the blurred test image shown in Fig. 7(b). Fig. 8(a) shows the output of the LUM using a 1-D horizontal window where $N = 7$

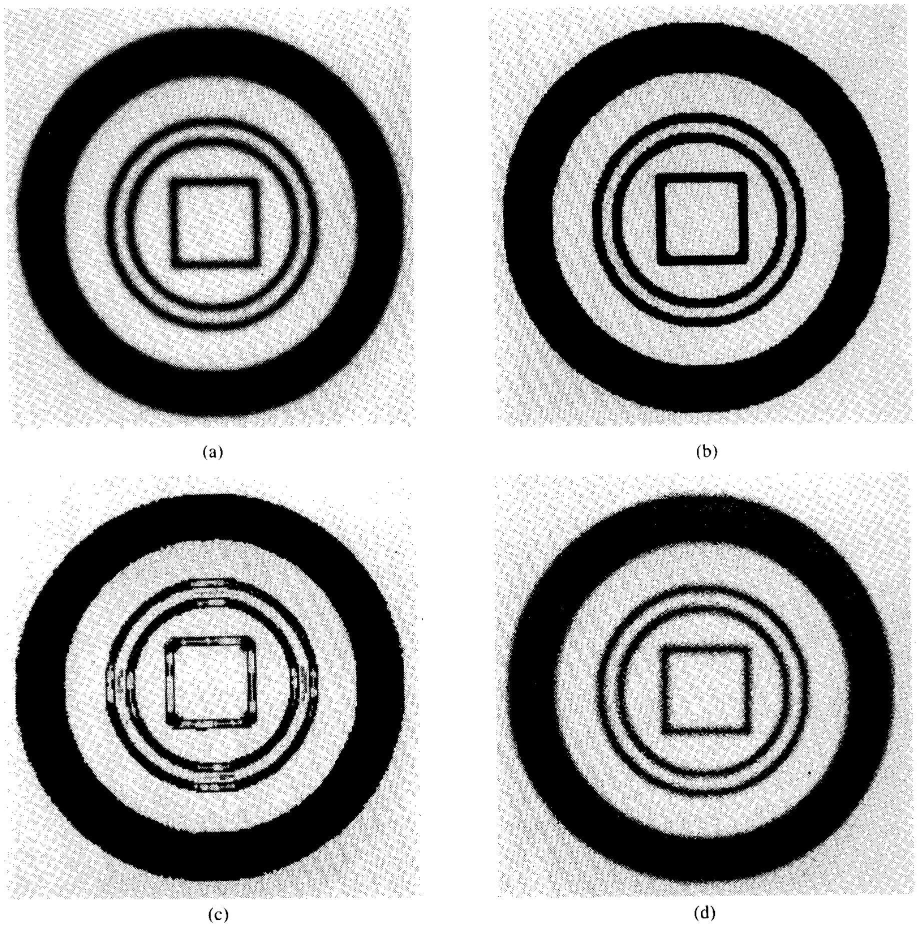


Fig. 9. Test image with Gaussian noise. (a) Noisy image. (b) Output of the LUM with a 7 × 7 square window where $k = 2$ and $l = 2$. (c) Output of the CS filter with a 7 × 7 square window where $j = 2$. (d) Output of unsharp masking where $s = 3$.

and $k = l = 1$ (maximum enhancement). This image illustrates how the LUM filter can be used to enhance images along a desired axis (in this case the horizontal axis). This type of directionally biased enhancement can be useful when combating a predominantly 1-D blurring function such as motion blurring. Fig. 8(b) shows the output of the CS filter. Here we have applied the CS filter to the image in two passes using a 1-D filter window. The image was first filtered with a horizontal window of length $N = 7$ where $j = 1$ and then with a vertical window of length $N = 7$ with $j = 1$. Note that this was the procedure used in [18]. Fig. 8(c) shows the output of the CS filter using a 7 × 7 square-window function. In both cases, the small-image features are distorted by the CS filter. However, the CS filter does a good job on the wide outer circle. Again, multiple passes with a smaller window would improve the detail-preserving characteristic of the CS filter. However, even with a window as small as 3 × 3, the CS filter would still remove any line one pixel wide.

For comparison we consider the unsharp masking technique. In two dimensions the output of this linear operator is given by

$$y(n_1, n_2) = x(n_1, n_2) + s\nabla. \tag{34}$$

The 2-D discrete Laplacian gradient is given by

$$\nabla = x(n_1, n_2) - \tfrac{1}{4}\{x(n_1 - 1, n_2) + x(n_1 + 1, n_2) + x(n_1, n_2 - 1) + x(n_1, n_2 + 1)\}. \tag{35}$$

Fig. 8(d) shows the output of the unsharp masking technique where $s = 3$. This technique does not fully restore the blurred edges and causes severe overshoot and undershoot. In general, unsharp masking will do a fairly good job visually sharpening an image in cases where the blurring is not severe.

Fig. 9 illustrates the LUM filter's performance in low levels of Gaussian noise. Fig. 9(a) shows the blurred test image of Fig. 6(b) corrupted with Gaussian noise having standard deviation 5. Fig. 9(b) shows the output of the LUM filter with a window size of 7 × 7 and parameters

TABLE III
MEAN ABSOLUTE ERROR FOR BLURRED TEST IMAGE IN GAUSSIAN NOISE

	Unfiltered	7 × 7 LUM	7 × 7 CS	Unsharp Masking
$\sigma = 0$	9.81	$(k = l = 1)$ 3.30	$(j = 1)$ 5.54	10.03
$\sigma = 2$	10.10	$(k = l = 1)$ 5.08	$(j = 1)$ 7.35	16.58
$\sigma = 5$	11.64	$(k = l = 2)$ 8.79	$(j = 2)$ 10.30	28.90
$\sigma = 10$	14.62	$(k = l = 5)$ 13.58	$(j = 5)$ 13.46	46.37

$k = 2$ and $l = 2$. Notice that the Gaussian noise present in the input image did not severely affect the performance of the LUM filter. Fig. 9(c) shows the output of the CS filter with a 7 × 7 square window where $j = 2$. In this case, the CS filter does a good job with respect to the additive noise; however, it still causes distortion in the thin-line features of the image. Finally, Fig. 9(d) shows the image filtered using unsharp masking with $s = 3$. This technique does a poor job in this case since there is substantial noise in the original image.

A quantitative comparison between the above-mentioned enhancing techniques is given in Table III. In general, the goals of image enhancement are based on highly subjective criteria making it difficult to quantify the performance of enhancement techniques. However, since we have restricted ourselves to the relatively straightforward task of restoring a blurred image, we can quantify the performance of these techniques by comparing their estimate to the original unblurred image. Table III shows that the LUM does well in restoring the blurred image, especially in low levels of additive Gaussian noise. Note that the CS filter slightly outperforms the LUM filter in high levels of Gaussian noise. This is because the CS filter does better at smoothing the constant regions that appear in this relatively simple test image. The LUM's performance can be improved in the higher noise levels by increasing k.

V. CONCLUSIONS

In this paper, we have presented a new class of rank-order-based filters, the LUM filters. Filters in this class are characterized by two parameters, one that adjusts smoothing and the other sharpening. In conjunction with the window size and shape, these two parameters allow the filter to exhibit a wide range of characteristics. These characteristics range from no smoothing to the median and from no sharpening to a maximal value. In addition, a variety of mixed smoothing and sharpening characteristics can be achieved.

In particular, we have demonstrated that the LUM filter can be designed to remove impulses with minimal image blurring. In comparison to other detail-preserving filters discussed in Section IV-B, the LUM filter performs extremely well.

In has also been demonstrated that the LUM can enhance edge gradients, even in the presence of noise. In Section IV-A the LUM filter was shown to be capable of removing impulsive-type noise while enhancing edges. In the enhancement experiments presented in Sections IV-A and IV-C, the LUM filter outperforms the CS filter and the unsharp masking technique.

In some applications, the user may wish to combine the LUM with other techniques. For instance, the LUM filter can make a useful front end to a linear edge detector [13]. The LUM filter can eliminate impulses and sharpen without introducing ringing. The linear edge detector will operate better on the cleaner image.

Because LUM filters perform well in a variety of applications and are so versatile, we believe they represent a very useful class. We also believe that LUM filters are simple to understand and design, making them very practical.

REFERENCES

[1] G. R. Arce and R. Foster, "Detail-preserving ranked-order based filters for image processing," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, no. 1, Jan. 1989.

[2] G. R. Arce and N. C. Gallagher, "State descriptions of the root signal set of median filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 30, no. 6, Dec. 1982.

[3] G. R. Arce, N. C. Gallagher, and T. A. Nodes, "Median filters: Theory for one and two dimensional filters," in *Image Enhancement and Restoration, Advances in Computer Vision and Image Processing*, vol. 2, T. S. Huang, Ed. Greenwich, CT: JAI Press, 1986, ch. 3, pp. 90-166.

[4] G. R. Arce and M. P. Mcloughlin, "Theoretical analysis of max/median filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-35, no. 1, Jan. 1987.

[5] J. B. Bednar and T. L. Watt, "Alpha-trimmed means and their relationship to median filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 32, pp. 145-152, Feb. 1984.

[6] C. G. Boncelet, "Some users of order statistic filtering in image compression," in *Proc. SPIE/SPSE Symp. Electron. Imaging Sci. Technol.*, (Santa Clara, CA), Feb. 1990.

[7] C. G. Boncelet, "Algorithms to compute order statistic distributions," *SIAM J. Sci. Comput.*, vol. 8, no. 5, Sept. 1987.

[8] A. 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. 31, pp. 1345-1349, Dec. 1983.

[9] E. Coyle and J.-H. Lin, "Stack filters and the mean absolute error criterion," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, no. 8, pp. 1244-1254, Aug. 1988.

[10] H. A. David, *Order Statistics*, 2nd ed. New York: Wiley, 1981.

[11] H. A. David and R. S. Mishriky, "Order statistics for discrete populations and for grouped samples," *J. Amer. Statis. Assoc.*, vol. 63, p. 1390, 1968.

[12] N. C. Gallagher and G. L. Wise, "A theoretical analysis of the properties of median filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 29, no. 6, Dec. 1981.

[13] R. C. Hardie, "Applications of nonlinear sharpeners to edge detection," in *Proc. 1992 Conf. Inform. Sci. Syst.*, (Princeton, NJ), Mar. 1992.

[14] P. Heinonon and Y. Neuvo, "FIR-median hybrid filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 35, pp. 832–838, June 1987.

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

[16] S.-J. Ko, Y. H. Lee, and M. K. Prasad, "Detail preserving weighted median filters," in *Proc. 1988 Conf. Inform. Sci. Syst.*, (Princeton, NJ), Mar. 1988.

[17] S.-J. Ko and Y. H. Lee, "Center weighted median filters and their applications to image enhancement," *IEEE Trans. Circuits Syst.*, vol. 38, no. 9, pp. 984–993, Sept. 1991.

[18] Y. H. Lee and A. T. Fam, "An edge gradient enhancing adaptive order statistic filter," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 35, May 1987.

[19] Y. H. Lee and S. A. Kassam, "Generalized median filtering and related non-linear filtering techniques," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 33, June 1985.

[20] Y. H. Lee and S. Tantaratana, "An adaptive edge enhancing order statistic filter," in *Proc. Int. Conf. Acoust., Speech, Signal Processing (ICASSP)*, New York, Apr. 1988.

[21] J. S. Jimmy Li, "A class of multi-shell min/max median filters," in *Proc. 1989 IEEE Int. Symp. Circuits Syst.*, (Oregon), May 1989.

[22] H. G. Longbotham and N. Barsalou, "The LMS, an adaptive optimal order statistic filter," in *Proc. SPIE/SPSE Symp. Electron. Imaging Sci. Technol.*, (Santa Clara, CA), Feb. 1990.

[23] H. G. Longbotham and N. Barsalou, "A class of generalized order statistic filters, similar to the median, that provide enhancement," in *Proc. SPIE/SPSE Symp. Electron. Imaging Sci. Technol.*, (San Jose, CA), Feb. 1991.

[24] C. L. Mallows, "Some theory of nonlinear smoothers," *Annals Statist.*, vol. 8, no. 4, pp. 695–715, 1980.

[25] P. A. Maragos and R. W. Schafer, "Morphological filters—Part I: Their set theoretic analysis and relations to linear shift invariant filters," and "Morphological filters—Part II: Their relations to median, order statistic and stack filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 35, no. 8, pp. 1153–1184, Aug. 1987.

[26] P. M. Narendra, "A separable median filter for image noise smoothing," *IEEE Trans. Pattern Analysis Mach. Intelligence*, vol. PAMI-3, no. 1, pp. 20–27, Jan. 1981.

[27] A. Nieminen, P. Heinonen, and Y. Neuvo, "A new class of detail-preserving filters for image processing," *IEEE Trans. Pattern Analysis Mach. Intelligence*, vol. PAMI-9, no. 1, pp. 74–90, Jan. 1987.

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

[29] K. Oflazer, "Design and implementation of a single chip 1-D median filter," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 31, no. 4, Oct. 1983.

[30] F. Palmieri and C. Boncelet, "Ll-filters: A new class of order statistic filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, no. 5, May 1989.

[31] I. Pitas and A. N. Venetsanopoulos, "Nonlinear mean filters in image processing," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 34, no. 3, June 1986.

[32] A. Restrepo and A. C. Bovik, "Adaptive trimmed mean filters for image restoration," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, no. 8, pp. 1326–1337, Aug. 1988.

[33] R. L. Stevenson and G. R. Arce, "Morphological filters: Statistics and further syntactical properties," *IEEE Trans. Circuits Syst.*, vol. CAS-34, no. 11, Nov. 1987.

[34] J. W. Tukey, "Nonseparable (nonsuperposable) methods for smoothing data," in *Cong. Rec. EASCON*, 1974, p. 673; also in *Exploratory Data Analysis*. Reading MA: Addison-Wesley, 1977.

[35] P. D. Wendt, E. J. Coyle, and N. C. Gallagher, "Stack filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 34, Aug. 1986.

Russell C. Hardie (S'91–M'92) was born in Baltimore, MD, on October 31, 1966. He received the B.S. degree from Loyola College, Baltimore, MD, in 1988 with high honors, the M.S. degree in electrical engineering from the University of Delaware, Newark, in 1990 and the Ph.D. degree in electrical engineering from the University of Delaware in 1992.

He is currently a Senior Scientist at Earth Satellite Corporation. His research interests include signal and image processing, multispectral image processing, pattern recognition, statistical communications, and 3-D cone beam tomography. He has also been a consultant to industry developing image processing algorithms and software.

Charles G. Boncelet (S'82–M'84) was born in New Jersey in 1958. He received the B.S. degree in applied and engineering physics from Cornell University, Ithaca, NY, in 1980 and the M.S. and Ph.D. degrees in electrical engineering and computer science from Princeton University, Princeton, NJ, in 1982 and 1984, respectively.

After working briefly for IBM and Bell Laboratories, he began as an Assistant Professor in Electrical Engineering at the University of Delaware in 1984. He was promoted to Associate Professor in 1989. He has served as a consultant to Princeton University and to the Alcoa Corporation. His research interests include image and signal processing, data compression, computer networks, and the theory of algorithms.

Dr. Boncelet is a member of Eta Kappa Nu, Tau Beta Pi, and SIAM, and has been active in SPIE.