Properties, implementations and applications of rank filters

R M Hodgson, D G Bailey, M J Naylor, A L M Ng and S J McNeill

The results of research on rank filters are presented. The relationship of rank filters with other filters is briefly discussed. The main properties of rank filters are listed and an explanation is given for these properties. Several software and hardware implementations of the filter are described. Major applications to image processing are discussed, including noise smoothing, cluster detection, skeletization, edge enhancement and edge detection.

Keywords: *rank filters, local filters, nonlinear filters*

A local operator is a filter whose output at a pixel is a function of the input values within the neighbourhood of that pixel[1]. This neighbourhood can be thought of as a window since, for each output pixel, only the pixel values within the window are used. The window is scanned across the input image, each position contributing to one pixel in the output image. The window can be of any shape, although it is almost always symmetrical about a centre point[2], and is usually square. Some common windows are shown in Figure 1. Local operators tend to have short calculation times since generally only a small number of input pixel values are operated on for each output pixel.

With linear filters, the output value is a linear combination or weighted average of the input pixel values from within the window. The choice of weights depends on the application of the filter. The theory of this class of filter is well understood[1]. However, linear filters are not suitable for many image processing tasks[3]. A linear low-pass filter is often used to reduce noise, but has the disadvantage that edges are blurred[2,4]. The use of a linear high-pass filter to sharpen or detect edges has the disadvantage that high-spatial-frequency noise is amplified[5,6] and spurious oscillations are produced near edges[5,6].

University of Canterbury, Private Bag, Christchurch, New Zealand

Nonlinear filters can be devised to overcome some of the disadvantages inherent in the use of linear filters[3,7]. The calculation time for local nonlinear filters is of the same order of magnitude as that for local linear filters since the same number of pixel values are processed. There are two broad classes of nonlinear filters[3]: those which are simple modifications of linear filters, and those which are not. Examples of filters from the first category are

- 'trimmed' filters
- 'gated' filters
- nonlinear combinations of linear filters

An example of a filter from the second category is the moment-based filter. Trimmed filters[8] are linear filters from which pixel values which are far removed in intensity from the central pixel or the median value of the window are excluded, reducing the noise sensitivity of the filter. Gated filters[3] use some function of the pixel values within the window to determine which of several linear filters will provide the resultant pixel value; eg a pixel is compared with the average of its neighbours and, if this difference is greater than a certain threshold, the centre pixel is replaced by the average[9]. An example of a nonlinear combination of

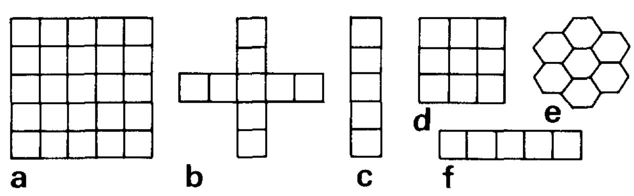

Figure 1. Commonly used windows: a, 5 × 5 square; b, 5 × 5 cross; c, 1 × 5 strip; d, 3 × 3 square; e, 3 × 3 hexagonal; f, 5 × 1 strip

linear filters is the Sobel filter[3,9], where the output is a nonlinear combination of the outputs of two linear edge detection filters. The moment-based filter uses the 'centre of gravity' or other moment of the pixel values within the window to detect edges[10].

A class of nonlinear filters of particular interest are the rank filters. With rank filters, all the pixel values within the window are ranked according to value, regardless of physical location within the window. The output of the filter is the pixel value selected from a specified position in this ranked list. Let the pixel values within a window of area N pixels be sorted into numerical order as

$$(f_1, f_2, f_3, \ldots, f_N) \tag{1}$$

where

$$f_1 \leqslant f_2 \leqslant \ldots f_N \tag{2}$$

The output is then selected as follows

$$\mathrm{rank}(i) = f_i \qquad 1 \leqslant i \leqslant N \tag{3}$$

When this is done for all possible window positions, it can be represented as

$$g = R_i(f) \tag{4}$$

where f is the input image, g is the processed image and i is the rank position selected. A special case of the rank filter when N is odd is the median filter, where the median rank position is selected. The other two special cases correspond to extreme rank position selection. These are called min and max filters[11] and are defined as follows

$$\min(f) = R_1(f) \tag{5a}$$

$$\max(f) = R_N(f) \tag{5b}$$

REVIEW OF PROPERTIES

A summary of important results follows. Most of the reported work on rank filters has concentrated on the median filter.

Property 1: Rank filters smooth noise[12,13]

The effect of applying a rank filter to a region of nominally constant intensity I and variance V is to reduce the variance. This effect can be observed in Figure 2, where the narrowing of the intensity histogram is indicative of the reduced variance. The high spatial frequencies associated with the noise are attenuated; in particular, oscillations in intensity with a period less than the window width will be smoothed[13]. In this case, since within the window there are pixel values at various stages in the cycle, pixels of approximately the same intensity are selected as the window moves across the oscillation. The exception to this is where rank positions near the median are used and the oscillation is binary or two valued. Then each of the two values is selected in different window positions and so the oscillation persists.

Filters that use rank positions near the median are especially useful for eliminating impulse noise[2,13,14]. This noise is usually caused by bit errors that occur during data capture or transmission. Since only a small proportion of pixels within a window are likely to be noise pixels and tend to occupy the extreme rank positions, these pixels will not be selected if rank positions near the median are used.

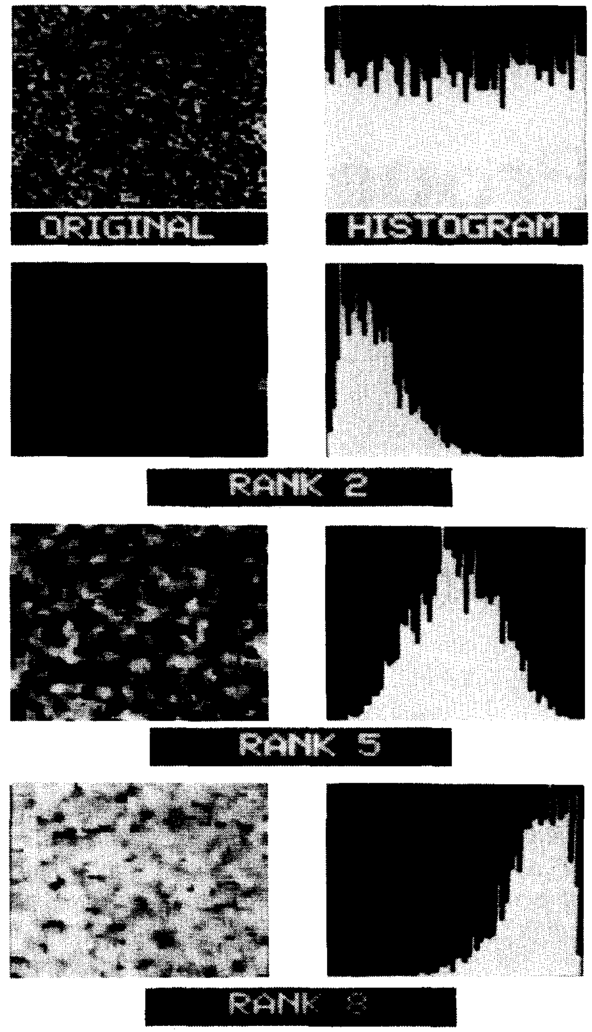

Figure 2. The effect of rank filters on a noise image using a 3×3 square window. The images and intensity histograms for ranks 2, 5 and 8 are shown

Much has been written on the noise smoothing properties of median filters[2,4,9,14–19] and many of the authors have provided statistical results.

Property 2: Application of a rank filter will change the mean intensity

The important exception to this generalization is the case that occurs when a median filter is used in a region where the noise distribution is symmetrical. In this case, the median filter does not change the mean intensity I in the region[12]. Selecting a rank position less than or greater than the median will reduce or increase I respectively. These effects are demonstrated in Figure 2, where the mean intensity can be seen to shift. With a symmetrical input distribution, the use of rank positions other than the median skews the output distribution, as shown in Figure 2.

The general effect of applying a rank filter to an image with a skewed noise distribution, when compared with the effect of the same filter on an image with a symmetrical noise distribution of the same mean and variance, is to shift the mean intensity in the direction of the skew. This shift can be explained as follows. When the mean value is calculated, outlying pixel values have a larger effect in determining the position of the mean than those closer in. When the median value is calculated, each pixel carries the same weight, and the median value is nearer the bulk of the intensity values than the mean value. This result also applies to other rank values near the median, but rank values near the extremes are determined mainly by the nature of the tails of the noise distribution.

Property 3: Rank filters preserve the shape of edges[2,20]

The shape of an intensity step, or ramp, between two adjacent regions of uniform but different intensities is preserved. Figure 3 shows the one-dimensional case. In general, this result carries over to two dimensions, as shown in Figure 4. When linear low-pass filters are applied to edges, step edges are blurred to ramps and the width of ramp edges is increased[4]. If the two regions are noisy, however, slight blurring does occur when a rank filter is applied[2,20]

To explain this effect a particular example will be used (see Figure 5). Consider a nine-element window positioned near an edge so that it contains five pixels from the region on one side of the edge and four from the region on the other side. The two regions have intensities of 110 and 80 respectively. The expected value of the median in this case is 110. If uniformly distributed noise is added to this edge, the pixel values are distributed uniformly between 100 and 120 on one side of the edge and between 70 and 90 on the other side. The median, corresponding to a rank position of 5, selects the minimum intensity of the five pixels between 100 and 120. This has an expected value of 103, whereas if there is no noise the expected value is 110. This is effectively a slight blurring of the edge.

Property 4: Application of a rank filter will shift the position of edges

Selecting a rank position less than or greater than the median will propagate the edge in the direction of the region of higher or lower intensity respectively[13,21] since the pixel values from one side of the edge will be selected while the window is still on the other side. This effect is illustrated in Figures 3 and 4.

Only the median filter will preserve the position of the edge if the window is symmetrical about its centre.

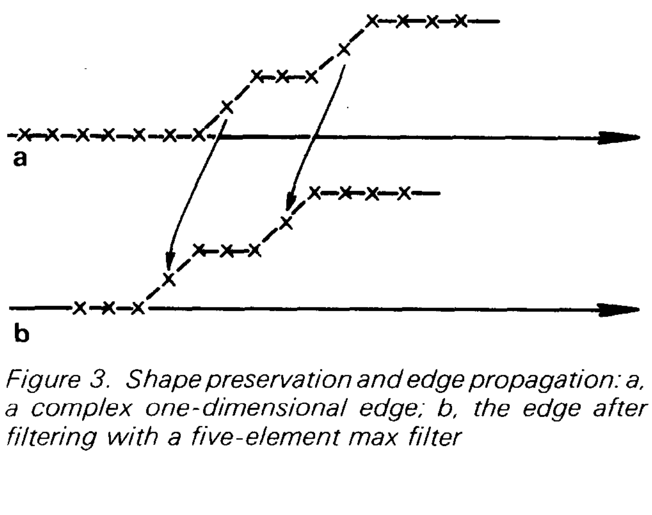

Figure 3. Shape preservation and edge propagation: a, a complex one-dimensional edge; b, the edge after filtering with a five-element max filter

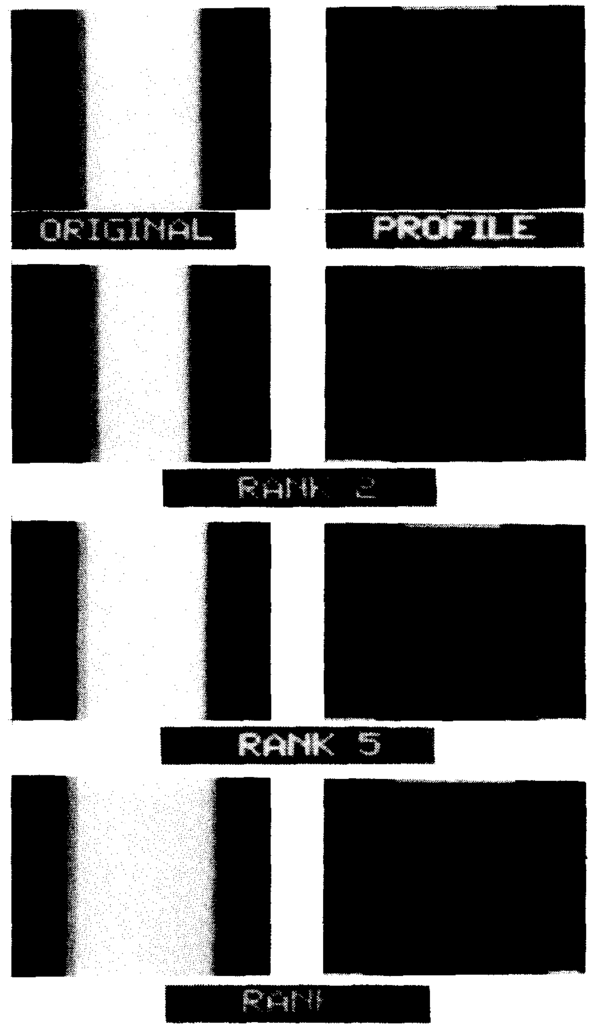

Figure 4. The effect of rank filters on edges, using a 9 × 1 strip window. The images and line profiles for ranks 2, 5 and 8 are shown

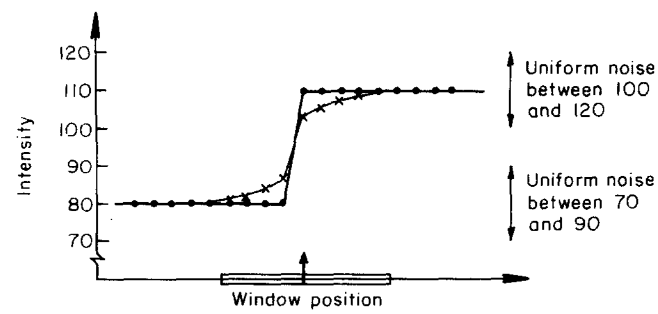

Figure 5. The blurring effect of noise on a one-dimensional edge subject to median filtering with a nine-element window: ●, no noise; ×, expected value for noisy data

This is because, when the window is on an edge, there are just as many points on one side of the centre pixel that are greater than the centre pixel value as there are on the other side (see Figure 6). Thus the median value is the value of the centre pixel, and no modification is made to the edge[13]. Any irregularities in the edge less than the width of the window will be smoothed by filters with rank positions close to the median.

Property 5: A rank filter will change all signals except a constant[13]

Repeated application of a rank filter will continue to change the image. This happens because the only fixed points for rank filters other than the median are images of uniform intensity. Thus, for rank filters in general, the only solution to the equation

$$f=R_i(f) \tag{6}$$

is

$$f=\text{constant} \tag{7}$$

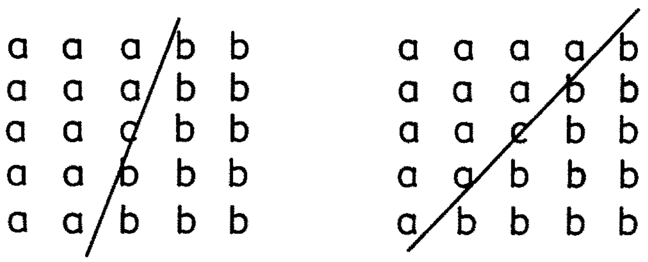

Figure 6. Straight edge preservation property of median filtering

This property is an extension of property 2 and is implicit in property 4. After any edges or intensity steps in the image have propagated to the edges of the image, there will be no edges left to propagate; therefore the image must be constant. With the median filter, edges do not propagate since edges are invariant; thus the fixed points of the median filter image consist of edges, regions of locally monotonic slope[8,22-25], and certain types of saddle point[23]. The fixed-point, or root, images for two different window shapes and sizes are shown in Figure 7.

Property 6: Min and max filters have specific properties[11,12]

The properties of min and max filters can be written as

$$\min(\max(\min(f)))=\min(f) \tag{8a}$$

$$\max(\min(\max(f)))=\max(f) \tag{8b}$$

$$\min(\max(f))\geqslant f\geqslant\max(\min(f)) \tag{9}$$

These properties are not held in common with the non min or max rank filters. However, for rank positions close to the extremes, equations (8) and (9) may be

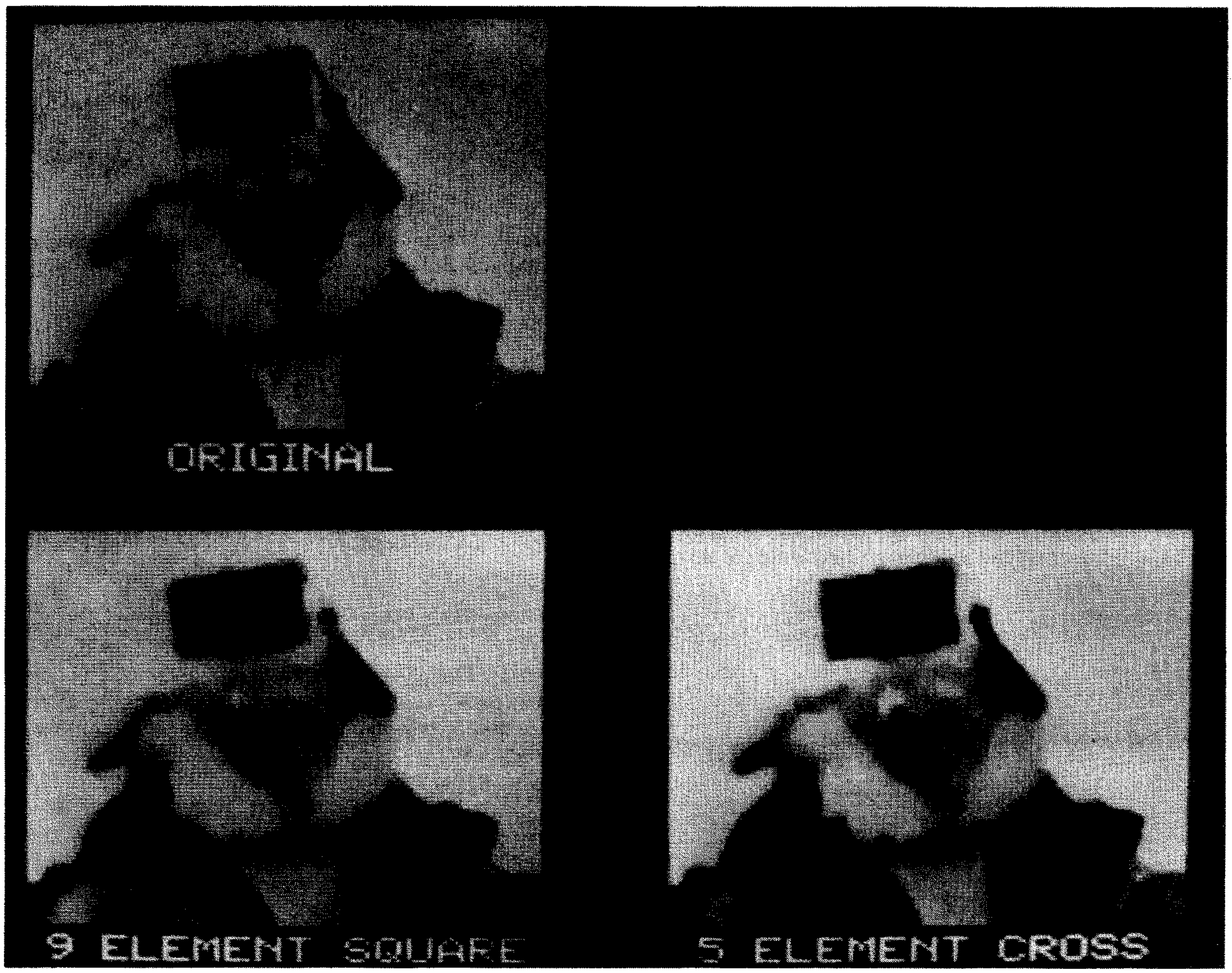

Figure 7. The fixed points of an image subject to median filtering. The lower left-hand image is the fixed-point image of a 3×3 square window, and the lower right-hand image is the fixed-point image of a 3×3 cross window

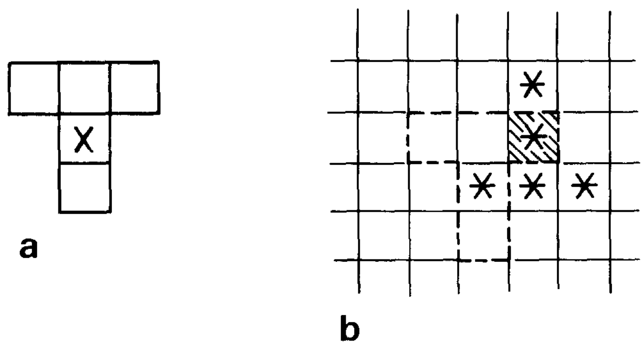

Figure 8. Local maximum from a max-filtered image: a, T-shaped filter; b, its effect on the local maximum before filtering (▧) and after filtering (∗); ---, one window position

considered to hold approximately, the approximation becoming less valid for positions towards the median.

Property 7: In a max-filtered image, all regions of local maxima contain the window rotated by 180°[12]

This property can be explained by considering the T-shaped window shown in Figure 8. If there is a pixel whose intensity is a local maximum (as determined by the window size and shape) then this pixel will be selected by the max filter when the window is in the positions marked by asterisks. The dotted lines show one such window position. In this way, the pixel which is a local maximum is selected from different positions from within the window, giving a local maximum the shape of the window rotated by 180°. A similar property holds for min filtering with regard to the local minima.

Property 8: Monotonically increasing functions of greyscale commute with rank filters[12]

Given a monotonically increasing function M defined as

$$g = M(f) \tag{10}$$

then

$$R_i(M(f)) = M(R_i(f)) \tag{11}$$

Since the order of the pixel values from within the window does not change, selecting the same rank position will select the same pixel. The function M may be applied either before or after rank filtering since M is a point operator, ie the resultant value at a pixel is independent of the value of neighbouring pixels. This property is of particular use in delaying some image processing decisions (eg selecting a threshold level) until the image is in a more suitable form (eg less noisy)[11].

An extension of this property is that any monotonically increasing function commutes with any series of rank filters[12], ie

$$R_i(R_j(R_k(M(f)))) = M(R_i(R_j(R_k(f)))) \tag{12}$$

Property 9: Complement property

Rank filtering a complemented image (ie an image which has its intensity range reversed) with a rank position i is equivalent to rank filtering the original image using the rank position $N+1-i$ and complementing the result[11,12]. Since complementing an image reverses the order of the ranked pixel values from within the window, the same pixel will be selected if the ranking is reversed. Thus

$$-f_{N+1-i} = (-f)_i \tag{13}$$

or when performed for the whole image

$$-R_{N+1-i}(f) = R_i(-f) \tag{14}$$

The consequence of this property and property 8 is that rank filters are commutative with monotonically decreasing functions of greyscale provided that the rank position used is modified appropriately.

Property 10: In the application of a rank filter, no new intensity values are generated

This property holds because at each window position one of the intensity values from within the window is selected for the output image. Thus the intensity values appearing in the output image will be a subset of those in the input image. This property is important when only a few of the available intensities appear in an image. With linear filters in general, new values are generated.

FAST METHODS

Several efficient methods have been devised for applying median filters to an image[8,26-28]. Two of these[26,27] use a histogram modification scheme to make use of the overlap from one window position to the next (only one pixel value changes in the one-dimensional case, while a small fraction of the window area changes in higher dimensions—see Figure 9). In this method, a histogram is taken of the window in the first position, and the median is derived from that. For subsequent window positions, the pixels which move out of the window are removed from the histogram and the new pixels are added. The median is then modified to represent the distribution in the new window position. This is repeated for all possible window positions. The method can be shown to be considerably faster than conventional sorting algorithms, especially for larger window sizes[27].

Bednar *et al.*[8] have reported a simplification of this method, using a sorted list rather than a histogram.

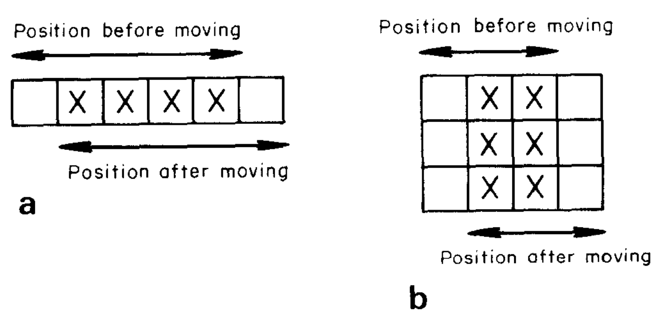

Figure 9. Overlap in the pixels sorted as the window moves: a, one-dimensional case (five-element window); b, two-dimensional case (3×3 window); X, overlapping or redundant pixels

Step	1	2	3	4	5
6=00110	0				
25=11001	1	1			
19=10011	1	0	0	1	1
16=10000	1	0	0	0	
23=10111	1	0	1		
8=01000	0				
20=10100	1	0	1		
30=11110	1	1			
15=01111	0				
MEDIAN	1	0	0	1	1
Value wanted	5th	2nd	2nd	2nd	1st
Number of 0s	3	4	2	1	0

Figure 10. Ataman's median sorting algorithm shown for nine elements

Their method is good for one-dimensional rank filtering, but in higher dimensions no real advantage is gained since several values in the sorted list need to be changed when moving from one window position to the next. If more than one rank position is required, this method has the advantage that all rank positions are readily available.

Another method[28] calculates the median value within a window by sorting the pixel values 1 bit at a time and discarding the values which will not be used. This is illustrated in the following example to find the median of nine 5-bit numbers (see Figure 10). In the first step the most significant bit is examined. Of the nine values, three begin with a '0'; therefore the median value begins with a '1' and those numbers beginning with a '0' are discarded. Of the six values that are left, the second is the median. The second most significant bit is now examined. Since four of the values that are left have a '0' and the second is required, the next bit in the median is a '0'. Values with a '1' are discarded. As this process is repeated for each bit, the median is built up 1 bit at a time. Although these methods are given for the median filter, they can be easily modified to perform rank filtering.

HARDWARE SYSTEMS FOR MEDIAN OR RANK FILTERING

Several hardware systems have been proposed or constructed for rank filtering. The earliest of these is an analogue circuit for selecting a particular ranked position from several voltages[29,30]. Although this circuit is fast since it uses parallel circuitry to perform the processing, it is impractical in digital image processing applications because of the inconvenience and limited accuracy of D/A conversion, analogue processing and A/D conversion. To select a different rank position, the circuitry needs to be significantly modified making this method unsuitable for general rank filtering.

Ataman *et al.*[28] have described an implementation of the median filter using the successive refinement method described earlier. The hardware they discussed should be able to perform rank filtering with only minor modifications. However, it can provide only one rank at a time. In applications where more than one rank position is required, as in the applications suggested by Bednar *et al.*[8] and Bovik *et al.*[31], this would be a disadvantage since it would require duplication or significant modification of the circuit. The main advantage of this method is the relatively low component count when compared with other possible discrete implementations.

Rank filtering is ideally suited to implementation by VLSI techniques since the filter can be designed using regular circuit structures[32]. Several methods have been proposed to apply VLSI techniques to one-dimensional median filtering[32-34]. These can be applied to two-dimensional processing using what is known as a 'separable median' filter[17,25]. With this filter, first the rows then the columns are median filtered, giving the median of medians (both one-dimensional operations). This is equivalent to median filtering with a horizontal strip window (eg as in Figure 1f) followed by median filtering with a vertical strip window (eg as in Figure 1c). Although this has similar properties to the conventional median filter[17,25] it cannot be easily extended to filters of arbitrary rank.

Fisher[35] has presented algorithms for one- and two-dimensional filtering. These methods use a linear pipeline of identical cells. Each cell has a stored value and receives a message action and a message value from the previous cell. The message action is performed and at the next cycle the message action and value are passed to the next cell. As the window moves, the values of the pixels which move out of the window are deleted from the sorted list, and the values of the pixels which move into the window are inserted. In this way the total overall processing is minimized, being spread efficiently among all the cells. The processing performed by each cell consists of comparing and swapping two intensity values. The area of silicon required to implement this algorithm is proportional to the number M of pixels within the window, an advantage over alternative algorithms which require an area of silicon proportional to M^2. For larger windows this property of the Fisher algorithm becomes a major advantage.

The authors of this paper have proposed a VLSI chip which will perform rank and other related filtering on a two-dimensional image using a 3×3 square window. This will use a conventional parallel bubble sorting algorithm, a method which becomes impractical for larger window sizes[32]. At the time of writing, the design for the prototype chip is nearing completion. This chip will be capable of processing images with pixel rates of the order of 100 ns per pixel, enabling images with up to 512 pixels per row to be processed at video rates.

APPLICATIONS OF RANK FILTERS

Application 1: Noise suppression

Rank and other nonlinear filters may be used with advantage for noise suppression where the information in the image being filtered is destroyed by conventional low-pass filtering. The filters mentioned in this section all have the property that to some extent they preserve edges in the original image.

The main application of the median filter is in noise suppression and much has been written in this field[2,4,9,15-20,36,37] The median filter works best on heavily tailed noise distributions (eg uniform distribution or exponential distribution)[2,31], and is very effective at removing spike noise[13,36].

Combinations of rank filters can also be used for noise suppression[8,11,21,31,38]. There are two main ways rank filters can be used

- sequential filter passes[11,21,38]
- in a weighted sum of values from different rank positions[8,31]

An algorithm that eliminates spike noise effectively using sequential passes is

$$g=\min(\max(\max(\min(f)))) \tag{15}$$

This can be considered as one of a family of filters. If we consider

$$\min_n(f)=\min(\min(\ldots\min(f))) \tag{16}$$

where the min filter is applied sequentially n times, then the following are all noise suppression filters[11,21]

$$g=\min_n(\max_n(f)) \tag{17a}$$

$$g=\max_n(\min_n(f)) \tag{17b}$$

$$g=\max_n(\min_{2n}(\max_n(f))) \tag{17c}$$

$$g=\min_n(\max_{2n}(\min_n(f))) \tag{17d}$$

The filters represented by equations (17a) and (17b) eliminate negative and positive going features respectively; the size of the feature eliminated depends on the parameter n. Rank positions other than the extremes give similar results. The main disadvantage of these filters is that, in general, the mean intensity level is not maintained if the input images are noisy. This can be inferred from property 6 above, which also suggests that the filters represented by equations (17c) and (17d) will be better in this respect.

A linear combination of rank filters also eliminates noise[8,31]. In general this type of filter may be represented as

$$g=\sum_{i=1}^{N} W_i R_i(f) \qquad \text{where} \quad \sum_{i=1}^{N} W_i=1 \tag{18}$$

Different schemes for selecting the weights give different families of filters which are optimal in a particular sense[31]. In both cases found in the literature, the weights are symmetrical about the median as in the equation

$$W_{N+1-i}=W_i \qquad 1\leqslant i\leqslant N/2 \tag{19}$$

Application 2: Low-pass, band-pass and high-pass filtering

Equations (17a)–(17d) represent a type of spatial low-pass filter, the cut frequency being dependent on n. In general, as n increases, the cut frequency decreases since larger features are eliminated. These are not true low-pass filters[12] since edges are preserved. When using these filters in two or higher dimensions, the frequency response is direction dependent, but this limitation can be overcome to a certain extent by using rank positions other than the extremes[38]. An example of this for a 3×3 window is

$$g=R_1(R_2(R_1(R_9(R_8(R_9(f)))))) \tag{20}$$

This corresponds to equation (17a) with $n=3$.

Since these filters are generically low pass in their response (apart from edge preservation), band-pass and high-pass filters may be derived[21,38] from them in the same way as for linear band-pass and high-pass filters[1]. An example of a band-pass filter is

$$g=\min_n(\max_{2n}(\min_n(f))) - \min_k(\max_{2k}(\min_k(f))) \qquad n<k \tag{21}$$

where the results from two low-pass filters with different cut frequencies are subtracted. When a low-pass-filtered image is subtracted from the original as in

$$g=f-\min_n(\max_{2n}(\min_n(f))) \tag{22}$$

a high-pass-filtered image is obtained. Similar equations may be derived for the low-pass filters given in equations (17a)–(17c). Low-pass-, band-pass- and high-pass-filtered images based on equation (17d) are shown in Figure 11.

Application 3: Shrinking and expanding, skeletizing

Min and max filters have been proposed as substitutes for shrink and expand operators when processing multivalued images[2,11,21]. Rank filters in general may be used, since they also propagate edges (property 2). Rank positions greater or less than the median will expand or shrink regions of high intensity respectively.

Several algorithms have been proposed for skeletizing images using min and max filters[21,39]. One of these[21] uses 'gated' min and max over several different window shapes and positions to extend the skeletization algorithms used with binary images. The other algorithm[39] uses min and max filters to shrink and expand the image in the following way. The image is shrunk using the iterated min filter for several different iteration lengths. Each of these is expanded once using the max filter, and subtracted from the previous shrunk image according to

$$g_n=\min_{n-1}(f)-\max(\min_n(f)) \tag{23}$$

This result is nonzero as a result of property 6; the points in the resultant image are those which are a distance $n-1$ from the edge of the object and are not adjacent to any points at a distance n from the edge. When the results for several n are summed the resultant image contains the skeleton of f

$$h=\sum_{n=1}^{m} g_n \tag{24}$$

These steps may be combined as follows to give what will be referred to as a 'skelet' filter. This filter combines the residue from one shrink/expand operation with the original image. Let

$$g=f-\max(\min(f)) \tag{25}$$

Then

$$\mathrm{skelet}(f)=\min(\max(\min(f)+g)) \tag{26}$$

Each time the skelet filter is iterated, the image is shrunk by a single layer of pixels, that layer being replaced by the skeleton. This process is repeated until no more change is made to the image (or until any changes made are insignificant). The fixed point of the skelet

Figure 11. Low-pass, band-pass and high-pass filtering of an image. The low-pass filter is described by equation (17d) with n=1*. The band-pass filter is described by equation (21) with* n=1 *and* k=2*. The high-pass filter is described by equation (22) with* n=2

filter is the skeleton of the image. Figure 12 shows an image after successive passes of the skelet filter.

Application 4: Streak and spot/cluster detection

Batchelor has used the min and max filters to detect spots and streaks in images, without detecting step edges[40] He uses the equation

$$g=f-\max(\min(f)) \qquad (27a)$$

to detect streaks and spots which have a higher intensity than the surrounding pixels, while use of the equation

$$g=\min(\max(f))-f \qquad (27b)$$

detects streaks and spots of lower intensity. The equation

$$g=\min(\max(f))-\max(\min(f)) \qquad (27c)$$

(being the sum of equations (27a) and (27b)) will detect both sets of streaks and spots. The size of the detected features can be increased by using the iterated min and max filters given by equation (16). After the filtering has been done, the resultant image g may be thresholded to obtain a feature map.

These equations can be used in a similar way to detect clusters[11]. Consider an image containing several small isolated regions of high intensity. The max filter is used repeatedly to expand these regions until some of them merge or fuse together. The min filter is then used to shrink the regions back to their original size. The regions which merged remain connected and may be detected by subtracting the original image. These interconnecting lines may be expanded again to select the original points that are clustered. This process is shown in Figure 13.

Application 5: Edge detection

If regions of interest, which are different in intensity from the surrounding pixels, are shrunk then the difference between the result and the original image will represent edge activity. This is because the edges between regions shift when the shrink operator is used, giving higher-intensity regions in the difference image where the edges have moved. Min and max filters as

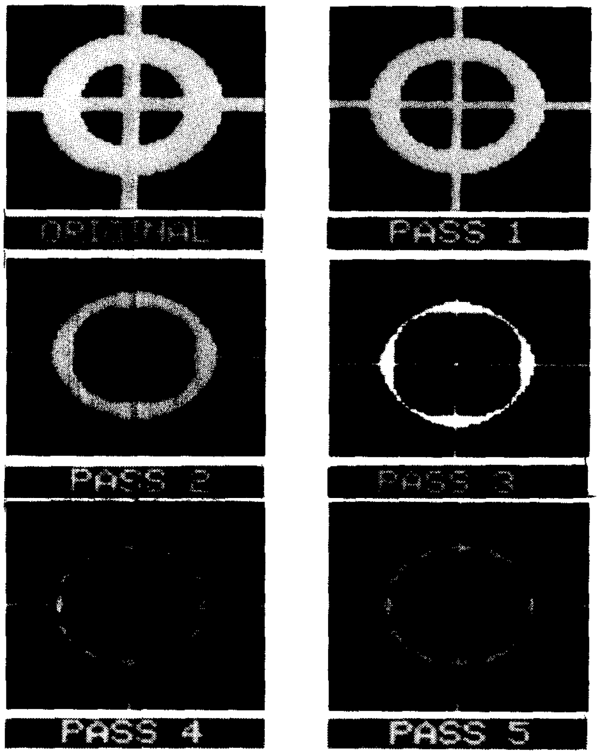

Figure 12. Successive passes of a skelet filter on a binary image. Note that, after pass 4, no further changes are made in the image

shown in the equations

$$g = f - \min(f) \tag{28a}$$

$$g = \max(f) - f \tag{28b}$$

have been used to do this[21,41] and the results are demonstrated in Figure 14. These filters may be generalized, using any rank positions, in the form

$$g = R_j(f) - R_i(f) \qquad i < j \tag{29}$$

The two rank filters are used to shrink or expand the regions of interest by differing amounts (property 4). The difference image then has higher intensity where the borders of the regions of interest have been shifted as demonstrated in one dimension in Figure 14. Because of the noise smoothing properties of rank filters (property 1) these filters are reasonably insensitive to noise, especially spike or other heavy-tailed noise.

When $j = N$ and $i = 1$ in equation (29), the filter gives the statistical range of the pixel values within the window; for other values of i and j a subrange is given. For this reason these filters are called range filters and are represented as

$$\mathrm{Ra}_{ji}(f) = R_j(f) - R_i(f) \tag{30}$$

This can be computed in a single pass of the window, rather than as a difference between two rank-filtered images, as follows

$$\mathrm{range}(j, i) = f_j - f_i \tag{31}$$

The result of range filtering an image with several values of i and j is shown in Figure 15. A companion paper describing the properties of this filter in more detail is in preparation[42].

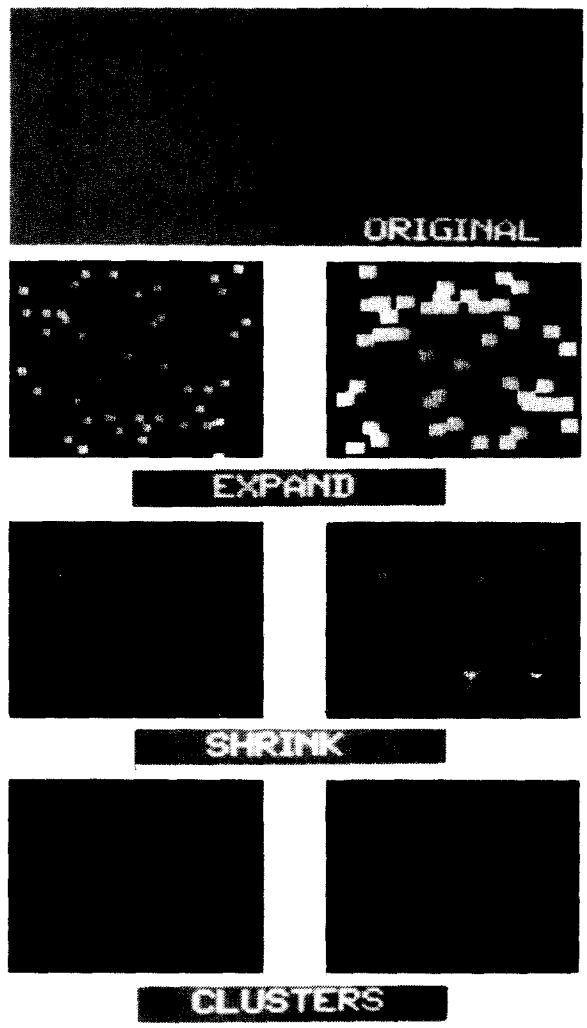

Figure 13. The use of rank filters for cluster detection. The second row shows one iteration of the max min filters and detects closely grouped clusters. The third row shows two iterations of the max min filters

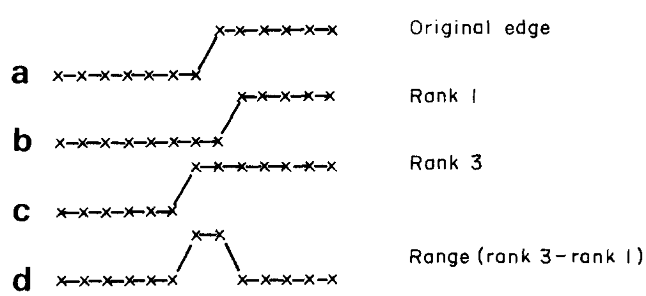

Figure 14. Edge detection by a range filter

Application 6: Edge enhancement

When an image is blurred, whether in the optical system forming the image, or as a result of low-pass filtering to remove noise, it is often desirable to sharpen or enhance the edges in the image. A gated rank filter may be used for this task. This filter compares the intensity of the centre pixel of the window with the mean of two rank values. If the centre pixel value is greater than the mean then the larger rank value is

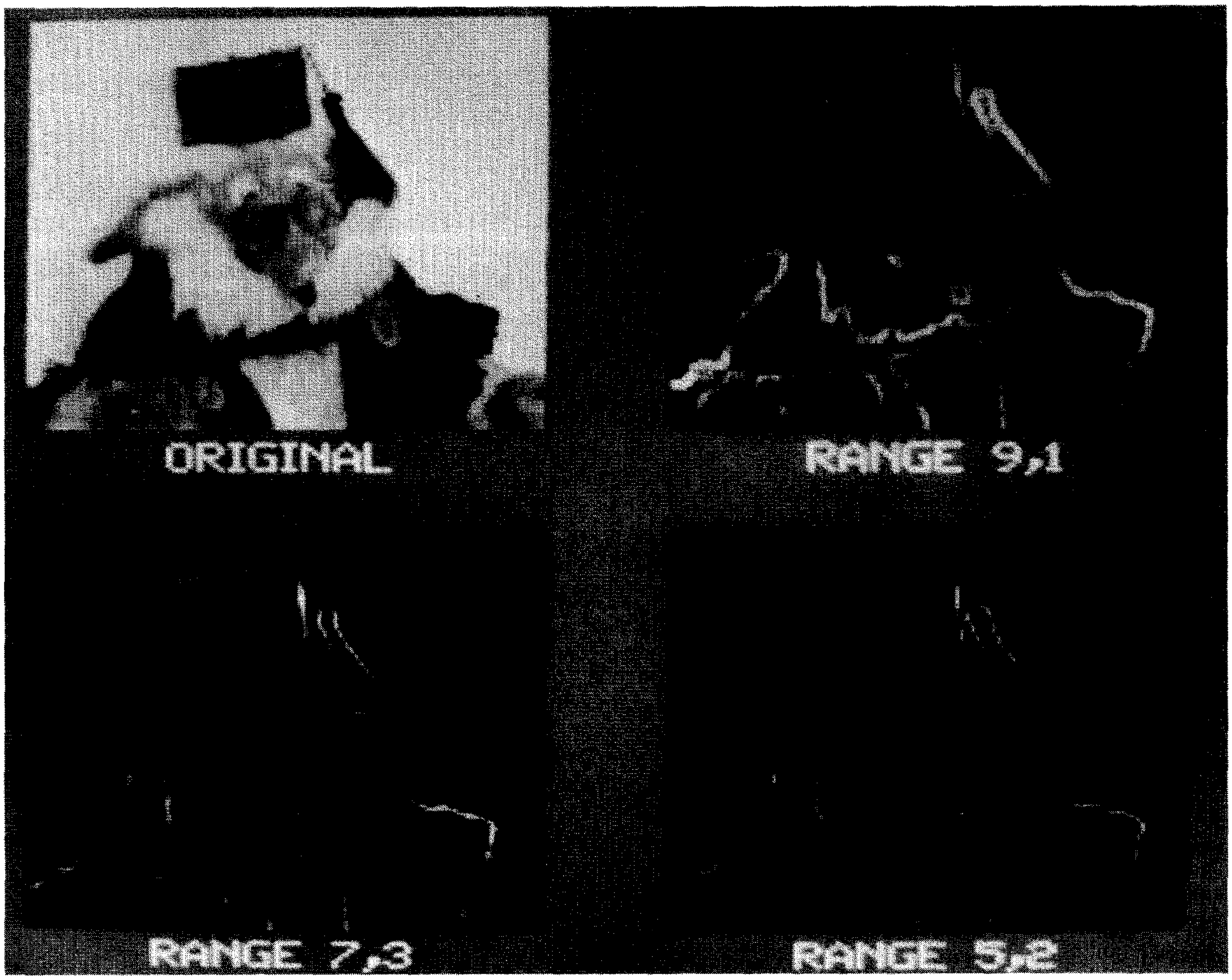

Figure 15. Edge detection using three different range filters

selected, otherwise the smaller is used. This may be represented as follows

$$\text{new value} = f_j \quad \text{if} \quad f_{\text{centre}} > (f_j + f_i)/2 \tag{32a}$$

$$\text{new value} = f_i \quad \text{if} \quad f_{\text{centre}} \leqslant (f_j + f_i)/2 \tag{32b}$$

An alternative viewpoint is to compare the centre pixel value with each of two rank values, selecting the nearer. This can be expressed as

$$\text{new value} = f_j \quad \text{if} \quad |f_i - f_{\text{centre}}| > |f_i - f_{\text{centre}}| \tag{33a}$$

$$\text{new value} = f_i \quad \text{if} \quad |f_j - f_{\text{centre}}| \leqslant |f_i - f_{\text{centre}}| \tag{33b}$$

When the window is in the vicinity of an edge, the two rank values are considered to represent the regions on either side of the edge. The centre pixel is assigned the value of the region that it is nearer to in intensity. It has been found that the use of extreme rank positions gives the best edge enhancing properties; however, such a filter also emphasizes the noise. Using rank positions somewhere between the median and the extreme positions is better in that noise enhancement is minimized. With this method of edge enhancement, ringing is prevented since no new pixel values are generated (property 10). Figure 16 shows the effect of this filter on several blurred and noisy edges.

SUMMARY AND CONCLUSIONS

In all the applications presented here rank filters perform as well as, or better than, conventional linear filters. When used for noise suppression, most of the edge information in the image is retained. When applied to streak and spot detection, features up to a desired size may be extracted, without detecting edges. For edge detection, range filters are less sensitive to noise than linear filters and perform about as well as the commonly used Sobel filter. With edge enhancement, the gated rank filter is less sensitive to noise than linear filters, and ringing is prevented.

The other applications or tasks discussed in this paper cannot be performed by conventional linear filters. The shrink and expand operators are restricted to binary image processing, and rank filters may be used as substitutes for these when processing greyscale images. Skeletization normally requires special-purpose algorithms but may be performed with

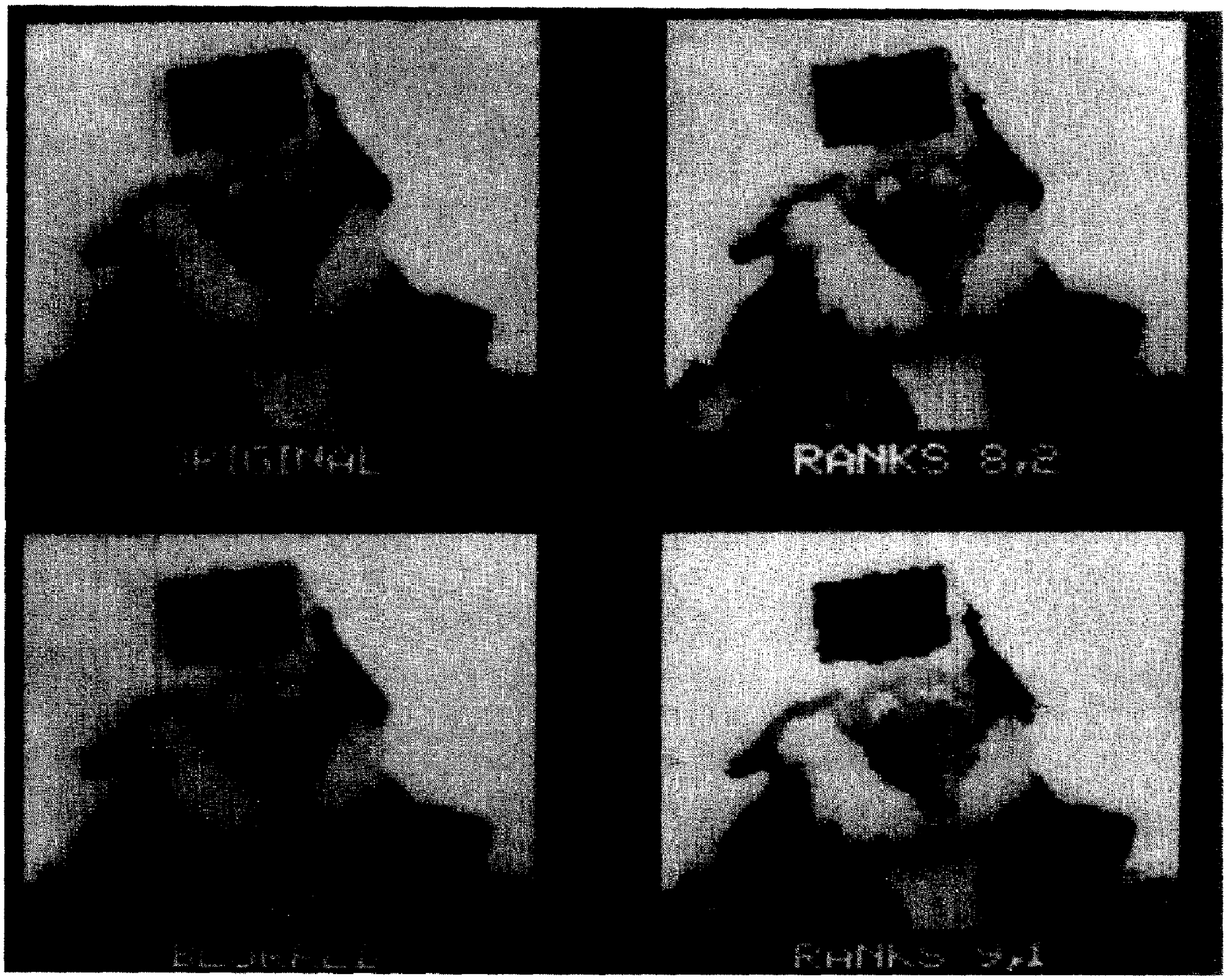

Figure 16. The use of the gated rank filter for edge enhancement. The top row shows the original image enhanced using ranks 8 and 2. The bottom row shows a blurred image enhanced using ranks 9 and 1

general-purpose rank filters at the expense of increased computation time.

Several implementation schemes for rank filtering, in both software and hardware, have been presented. No real comparison has been made between the methods since there is not enough published data for a complete assessment of the merits of each scheme.

Overall it has been shown that rank filters have many properties that make them a useful tool in image processing. The general nature of this tool has been shown through its application to a wide variety of tasks.

ACKNOWLEDGEMENTS

D G Bailey, M J Naylor and S J McNeill are supported by University Grants Committee postgraduate scholarships. A L M Ng is supported by a grant from the Forest Research Institute and a Christchurch City Council postgraduate scholarship.

REFERENCES

1 **Castleman, K R** *Digital image processing* Prentice-Hall, Englewood Cliffs, NJ, USA (1979)

2 **Justusson, B I** 'Median filtering: statistical properties' in **Huang, T S** (ed.) *Two dimensional digital signal processing II, Top. Appl. Phys.* Vol 43 (1981) pp 161–196

3 **Bailey, D G, Hodgson, R M and McNeill, S J** 'Local filters in digital image processing' *Proc. 21st Natl Electronics Conf., Christchurch, New Zealand 22–24 August 1984* pp 95–100

4 **Chin, R T and Yeh, C L** 'Quantitative evaluation of some edge preserving noise smoothing techniques' *Comput. Vision, Graphics, Image Process.* Vol 23 (1983) pp 67–91

5 **Frieden, B R** 'Image restoration by discrete convolution of minimal length' *J. Opt. Soc. Am.* Vol 64 (1974) pp 682–686

6 **Frieden, B R** 'A new restoring algorithm for the preferential enhancement of edge gradients' *J. Opt. Soc. Am* Vol 66 (1976) pp 280–283

7 **Frieden, B R** 'Image enhancement and restoration' in **Huang, T S** (ed.) *Picture processing and digital filtering, Top. Appl. Phys.* Vol 6 (1975) pp 179–248

8 **Bednar, J B and Watt, T L** 'Alpha trimmed means and their relationship to median filters' *IEEE Trans. Acoust., Speech, Signal Process.* Vol 32 (1984) pp 145–153

9 **Pratt, W K** *Digital image processing* Wiley, New York, USA (1978)

10 **Suciu, R E and Reeves, A P** 'A comparison of differential and moment based edge detectors' *Proc. Conf. on Pattern Recognition and Image Processing* (1982) pp 97–102

11 **Nakagawa, Y and Rosenfeld, A** 'A note on the use of local MIN and MAX operations in digital picture processing' *IEEE Trans. Syst., Man, Cybern.* Vol 8 (1978) pp 632–635

12 **Heygster, G** 'Rank filters in digital image processing' *Comput. Graphics Image Process.* Vol 19 (1982) pp 148–164

13 **Nodes, T A and Gallagher, N C** 'Median filters: some modifications and their properties' *IEEE Trans. Acoust., Speech, Signal Process.* Vol 30 (1982) pp 739–746

14 **Ataman, E, Aatre, V K and Wong, K M** 'Some statistical properties of median filters' *IEEE Trans. Acoust., Speech, Signal Process.* Vol 29 (1981) pp 1073–1075

15 **Velleman, P F** 'Robust non-linear data smoothers: definitions and recommendations' *Proc. Natl Acad. Sci. USA* Vol 74 (1977) pp 434–436

16 **Tukey, J W** *Exploratory data analysis* Addison-Wesley, Reading, MA, USA (1977)

17 **Narendra, P M** 'A separable median filter for image noise smoothing' *IEEE Trans. Pattern Anal. Mach. Intell.* Vol 3 (1980) pp 20–29

18 **Frieden, B R** 'Some statistical properties of the median window' in **Rhodes, W T, Fienup, J R and Saleh, B E A** (eds) *Transformations in optical signal processing, SPIE* Vol 373 (1981) pp 219–224

19 **Kuhlman, F and Wise, G L** 'On second moment properties of median filtered sequences of independent data' *IEEE Trans. Commun.* Vol 29 (1981) pp 1374–1379

20 **Yang, G J and Huang, T S** 'The effect of median filtering on edge location estimation' *Comput. Graphics Image Process.* Vol 15 (1981) pp 224–245

21 **Goetcherian, V** 'From binary to grey tone image processing using fuzzy logic concepts' *Pattern Recognition* Vol 12 (1980) pp 7–15

22 **Gallagher, N C and Wise, G L** 'A theoretical analysis of the properties of median filters' *IEEE Trans. Acoust., Speech, Signal Process.* Vol 29 (1981) pp 1136–1141

23 **Tyan, S G** 'Median filtering: deterministic properties' in **Huang, T S** (ed.) *Two dimensional digital signal processing II, Top. Appl. Phys.* Vol 43 (1981) pp 197–217

24 **Arce, G A and Gallagher, N C** 'State description for the root signal set of median filters' *IEEE Trans. Acoust., Speech, Signal Process.* Vol 30 (1982) pp 894–902

25 **Nodes, T A and Gallagher, N C** 'Two dimensional root structures and convergence properties of the separable median filter' *IEEE Trans. Acoust., Speech, Signal Process.* Vol 31 (1983) pp 1350–1365

26 **Garibotto, G and Lambarelli, L** 'Fast on-line implementation of two dimensional median filtering' *Electron. Lett.* Vol 15 (January 1979) pp 24–25

27 **Huang, T S, Yang, G Y and Tang, G Y** 'A fast two dimensional median filtering algorithm' *IEEE Trans. Acoust., Speech, Signal Process.* Vol 27 (1979) pp 13–18

28 **Ataman, E, Aatre, V K and Wong, K M** 'A fast method for real time median filtering' *IEEE Trans. Acoust., Speech, Signal Process.* Vol 28 (1980) pp 415–420

29 **Morgan, D R** 'Analogue sorting network ranks inputs by amplitude and allows selection' *Electron. Des.* Vol 21 No 2 (18 January 1973) pp 72–74

30 **Morgan, D R** 'Correction to analogue rank sorting network' *Electron. Des.* Vol 21 No 17 (16 August 1973) p 7

31 **Bovik, A C, Huang, T S and Munson, D C** 'A generalisation of median filtering using linear combinations of order statistics' *IEEE Trans. Acoust., Speech, Signal Process.* Vol 31 (1983) pp 1342–1349

32 **Oflazer, K** 'Design and implementation of a single chip 1d median filter' *IEEE Trans. Acoust., Speech, Signal Process.* Vol 31 (1983) pp 1164–1168

33 **Eversole, W L, Mayer, D J, Frazee, F B and Cheek, T F, Jr** 'Investigation of VLSI technologies for image processing' *Proc. ARPA Workshop on Image Understanding* (November 1978) pp 191–195

34 **Shamos, M I** 'Robust picture processing operators and their implementation as circuits' *Proc. ARPA Workshop on Image Understanding* (November 1978) pp 127–129

35 **Fisher, A L** 'Systolic algorithms for running order statistics in signal and image processing' *J. Digital Syst.* Vol 6 (1982) pp 251–264

36 **Rabiner, L R, Sambar, M R and Schmidt, C E** 'Applications of a nonlinear smoothing algorithm to speech processing' *IEEE Trans. Acoust., Speech, Signal Process.* Vol 23 (1975) pp 552–557

37 **Jayant, N S** 'Average and median based smoothing techniques for improving digital speech quality in the presence of transmission errors' *IEEE Trans. Commun.* Vol 24 (1976) pp 1043–1045

38 **Preston, K** 'Xi filters' *IEEE Trans. Acoust., Speech, Signal Process.* Vol 31 (1983) pp 861–876

39 **Peleg, S and Rosenfeld, A** 'A MIN–MAX medial axis transformation' *IEEE Trans. Pattern Anal. Mach. Intell.* Vol 3 (1981) pp 208–210

40 **Batchelor, B G** 'Streak and spot detection in digital pictures' *Electron. Lett.* Vol 15 (1979) pp 352–353

41 **Pal, S K and King, R A** 'On edge detection of X-ray images using fuzzy sets' *IEEE Trans. Pattern Anal. Mach. Intell.* Vol 5 (1983) pp 69–77

42 **Bailey, D G and Hodgson, R M** 'Range filters: local intensity subrange filters and their properties' in preparation