Image Processing and Computer Vision

Robert M. Haralick Editor

The Weighted Median Filter

D. R. K. BROWNRIGG

ABSTRACT: The median filter is well-known [1, 2]. *However, if a user wishes to predefine a set of feature types to remove or retain, the median filter does not necessarily satisfy the requirements. A more general filter, called the Weighted Median Filter, of which the median filter is a special case, is described. It enables filters to be designed with a wide variety of properties. Particular cases of filter requirements are discussed and the corresponding filters are derived. The notion of a minimal weighted median filter, of a subclass that act identically, is introduced and discussed. The question of finding the number of distinct ways a class of filters can act is considered and solved for some classes.*

1. INTRODUCTION

The one-dimensional median filter was devised by Tukey [1]. Some discussion of it, and an extension to two dimensions, is given by Pratt [2].

For a raster image, the procedure is to take a number of values in the neighborhood of a pixel, find their median, and use this to replace the value of the pixel.

The effect is to remove energy from the image as high and low data values, compared to the surroundings, are removed. The filter may operate over a large extent to remove objects of size less than the extent. Thus, for example, stars may be removed from a raster image (digitized photograph or CCD image) to provide a background level that may then be subtracted from the original image before performing photometry.

Hence, the median filter may be used for removing spike noise from an image or backgrounding it. However, some undesirable effects may arise if due care is not taken, and on examination, it may be found that the median filter cannot be made to have the desired effect.

Consider the filter skeleton

$$\begin{matrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{matrix} \qquad (1)$$

This indicates that the values to be taken are the data value itself, together with its side, top, and bottom neighbors for a two-dimensional image. The five values are sorted and the median replaces the center value. Two basic requirements that might be made are

1. One pixel width high/low streaks, corresponding to emulsion scratches, or in a CCD image, a saturated pixel overspill in a lane, should be removed.
2. Any rectangular block of differing values, such as an intrusive intensity-scaling stepwedge, should remain unaffected.

In passing, it may be noted that averaging filters, such as the Martin Filter [3, 4] smear edges and corners of such blocks into the image and thus, do not satisfy Condition 2. However, Filter (1) will preserve high/low value blocks. For instance, consider the data

$$\begin{array}{cccc} 100 & 100 & 8 & 9 \\ 100 & 100 & 12 & 10 \\ 100 & 100\ (A) & 3 & 5 \\ 6 & 5 & 4 & 7 \\ 7 & 2 & 3 & 6 \end{array} \qquad (i)$$

Filter (1) acting at (A) obtains the (sorted) list of values (3, 5, 100, 100, 100) of which the median is 100 and the (high) intrusive block is unchanged.

Condition 1 will not always be met, however. A high transmission lane due to diagonal scratching is dealt with. For instance, for data

$$\begin{array}{ccccc} 100 & 4 & 7 & 14 & 6 \\ 7 & 100\ (A) & 10 & 11 & 5 \\ 8 & 12 & 100 & 9 & 8 \end{array} \qquad (ii)$$

at point (A), Filter (1) gives (4, 7, 10, 12, 100) with median 10, and the scratch is removed.

However, for vertical or horizontal scratches, or CCD streaks, such as in

$$\begin{array}{ccc} 7 & 100 & 8 \\ 5 & 100\ (A) & 12 \\ 9 & 100 & 3 \end{array} \qquad (iii)$$

at point (A), Filter (1) gives (5, 12, 100, 100, 100), median 100, and the defect is not removed.

There are other forms of median filter that can be tried, such as

$$\begin{array}{ccc} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{array} \qquad (2)$$

This indicates that the data value and its eight close neighbor values should be sorted and the median taken. With data (*ii*) and (*iii*), this provides sorted lists of (4, 7, 7, 8, 10, 12, 100, 100, 100), median 10, and (3, 5, 7, 8, 9, 12, 100, 100, 100), median 9, respectively.

That is, a high/low pixel line is removed irrespective of direction, thus satisfying Condition 1. Condition 2 is violated, though, as Filter (2) at point (A) of data (*i*) gives the sorted list (3, 4, 5, 6, 12, 100, 100, 100, 100), median 12, so that the block corner is not preserved.

Clearly, some predictive approach is needed for the type of filter required. This is developed below (Section 3), where it is shown that any simple median filter of the same type as (1) or (2) cannot satisfy both Conditions 1 and 2.

This motivates the introduction of filters that have the desirable spike noise removal and backgrounding capabilities of the median filter, but have greater flexibility in specification to allow for control in the way that they operate.

The Weighted Median Filter has this flexibility.

2. THE WEIGHTED MEDIAN FILTER

We define a two-dimensional (2D) weighted median filter (WMF) of extent $2n + 1$ to be the array of coefficients: $\{a(i, j)$: $-n \le i, j \le n$: $a(i, j)$ nonnegative integers: $\mathrm{sum}(a(i, j)$; $i, j = -n \ldots n)$ odd integer$\}$.

The operation of the filter at point (s, t) of a data array D is to take $a(i, j)$ copies of $D(s + i, t + j)$ for $i, j = -n, \ldots, n$, a total of $S = \mathrm{sum}(a(i, j);\ i, j = -n, \ldots, n)$ values where S is odd. These are sorted into ascending order ($L(k)$; $k = 1, \ldots, S$) and the median M is taken, that is, $M = L((S + 1)/2)$.

If $|(M - D(s, t))| > T$, where T is a given threshold value, possibly zero, then $C(s, t) = M$; otherwise $C(s, t) = D(s, t)$, where C is the modified image. (Note that C is used rather than replacing directly in D so as to avoid asymmetries and propagation effects in the filtered image that would then depend on the direction the filter is moved across the data.)

Logically, C is then copied into D, and in backgrounding, the process may be repeated as long as some values of D are replaced. In practice, if the filter is reapplied, C is the input image and D the result image, or for greater space efficiency C may be replaced with a rolling buffer of width equal to the filter extent and length equal to the image length. It should be noted that it is possible to have distributions where some filters lead to a loop occurring where pairs of values are interchanged at each step. This is discussed in Section 6.

One-dimensional filters may be obtained by setting the appropriate elements of a to 0. For example, a horizontal filter is obtained by setting $a(i, j) = 0$ for $i \neq 0$.

Further, three- or higher-dimensional filtering can be applied by a simple extension of the number of indices. Also, asymmetric filters may be devised if required, such as

$$\begin{array}{ccc} 0 & 0 & 0 \\ 0 & 2 & 1 \\ 0 & 1 & 1 \end{array}$$

Note that S is odd to provide a central value. If S even is allowed, an off-center value has to be chosen and this affects the symmetry of likelihood of selection of high or low values. For example, consider the filter

$$\begin{array}{ccc} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{array}$$

Denoting the occurrence of the central value of the filter (CVF) in the sorted list by bracketing its occurrence value, the possible orderings are

(2), 1, 1, 1, 1, 1, 1, 1, 1
1, (2), 1, 1, 1, 1, 1, 1, 1
1, 1, (2), 1, 1, 1, 1, 1, 1
1, 1, 1, (2), 1, 1, 1, 1, 1
1, 1, 1, 1, (2), 1, 1, 1, 1
1, 1, 1, 1, 1, (2), 1, 1, 1
1, 1, 1, 1, 1, 1, (2), 1, 1
1, 1, 1, 1, 1, 1, 1, (2), 1
1, 1, 1, 1, 1, 1, 1, 1, (2)

Now the CVF provides the median in two cases; when there are three values smaller and three larger; and when there are four smaller and four larger (if the median is taken as at $L/2$). This constitutes an asymmetry as five smaller and three larger does not give the CVF as the median.

One example of the effect is in block-corner preservation. A low-value block in a high background will be retained, but a high-value block in a low background will be affected. For example

$$\begin{matrix} 00000 \\ 00111 \\ 00111 \\ 00111 \end{matrix} \rightarrow \begin{matrix} 00000 \\ 00011 \\ 00111 \\ 00111 \end{matrix} \quad \text{but} \quad \begin{matrix} 11111 \\ 11000 \\ 11000 \\ 11000 \end{matrix} \rightarrow \begin{matrix} 11111 \\ 11000 \\ 11000 \\ 11000 \end{matrix}$$

If S is limited to being odd, this anomaly does not arise.

To avoid biasing the replacement to the top, bottom, or one side of a data value (though it may be done if required), a natural limitation on filter values for general use is

$$\begin{aligned} a(i, j) &= a(-i, -j) = a(-i, j) = a(i, -j) \\ &= a(j, i) = a(-j, -i) = a(-j, i) = a(j, -i) \end{aligned}$$

This gives eightfold symmetry for the square array. For a filter of extent three, this gives the form

$$\begin{matrix} r & s & r \\ s & t & s \\ r & s & r \end{matrix}$$

and for a filter of extent five, a form of

$$\begin{matrix} u & v & w & v & u \\ v & r & s & r & v \\ w & s & t & s & w \\ v & r & s & r & v \\ u & v & w & v & u \end{matrix}$$

Note that this restriction is not necessary for use of the filter. Sometimes asymmetric filters will be useful and desirable as when dealing with edge and corner values in an image.

A WMF may operate at any spacing on the data. For example, the filter

$$\begin{matrix} 0 & 1 & 0 \\ 1 & 3 & 1 \\ 0 & 1 & 0 \end{matrix}$$

of extent three, operating with spacing two on the data (*iv*), will pick out the bracketed values in operating at point *A*.

$$\begin{matrix} 14 & 52 & (36) & 22 & 11 \\ 18 & 17 & 42 & 33 & 12 \\ (21) & 31 & (36)\ (A) & 18 & (\ 9) \\ 14 & 15 & 27 & 19 & 21 \\ 35 & 24 & (36) & 12 & 23 \end{matrix} \qquad (iv)$$

This is, clearly, equivalent to the filter of extent five

$$\begin{matrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 3 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{matrix}$$

operating with spacing one. Thus, the concept of spacing adds nothing, logically, to the capabilities of WMFs, but aids greatly in compressing filter descriptions (nine numbers instead of 25 above).

The spacing setting allows backgrounding to be performed with a filter of only a few values. That is, a filter of numerical extent $2n + 1$ applied with spacing m is equivalent to some filter of extent $2nm + 1$ and may remove objects of up to size $2nm$. Thus, objects (such as stars in an astronomical image) extending over many pixels may be removed in backgrounding.

As stated above, asymmetric filters are useful in simple handling of edges and corners of a raster image. For example, there is no clear way to handle the data

$$\begin{array}{|lll} \hline 100\ (A) & 12 & 14 \\ 16 & 22 & 33 \\ 9 & 14 & 25 \end{array}$$

at the corner with a filter such as

$$\begin{matrix} 0 & 1 & 0 \\ 1 & 3 & 1 \\ 0 & 1 & 0 \end{matrix}$$

If edge replication is used, a high noise value at *A* will not be removed. That is, replication gives

$$\begin{array}{r|lll} 100 & 100 & 12 & 14 \\ \hline 100 & 100\ (A) & 12 & 14 \\ 16 & 16 & 22 & 33 \\ 9 & 9 & 14 & 25 \end{array}$$

leading to a sorted list (12, 16, 100, 100, 100, 100, 100), median 100. The corner value has been made to look like a corner of an intrusive block. However, different asymmetric WMFs can be specified for each corner, such as

$$\begin{matrix} 0 & 0 & 0 \\ 0 & 2 & 1 \\ 0 & 1 & 1 \end{matrix}$$

for the top left, and then this filter gives a sorted list (12, 16, 22, 100, 100), median 22, and corner noise spikes are removed.

3. THE DESIGN AND USE OF A WMF OF EXTENT THREE FOR AN APPLICATION IN ASTRONOMICAL SURVEY PHOTOGRAPHY

Plate I shows a section of SERC Atlas Field 116, digitized as an array of 450 × 450 pixels with intensity scaling (0, 172) for black to white. On the left is a block of stepwedges used for intensity calibration. Running across the lower half of the photograph are two plate defects arising from the passage of sun-illuminated satellites over the field.

It is desirable to process the image to remove the satellite trails while leaving the calibration blocks unaffected. These conditions may be expressed as

1. Removal of high/low streaks.
2. Block preservation for blocks of any orientation and size greater than the filter extent.

Consider a WMF of extent three

$$\begin{array}{ccc} r & s & r \\ s & t & s \\ r & s & r \end{array}$$

so $t > 0$ and odd integer; $r, s \geq 0$ and integer.

For blocks to be preserved, the corners must be preserved. Thus, in applying a WMF at a corner value, the number of values from inside the block must exceed that of values outside the block. For a block with edges aligned with the data axes, at a corner

$$\begin{array}{c|cc} r & s & r \\ \hline s & t & s \\ r & s & r \end{array}$$

The requirement is that

$$t + 2s + r > 2s + 3r$$

that is,

$$t > 2r$$

For a block at an angle to the data axes, such as

$$\begin{array}{ccc} r & s & r \\ s & t & s \\ r & s & r \end{array}$$

the requirement is

$$t + 2r + s > 2r + 3s$$

$$t > 2s$$

To remove all line defects, including those with bends in them, it is necessary that the number of values in the line is less than that off the line when the filter is acting at a point on the line. There are three sets of cases. First, for a vertical or horizontal line or at the right angle bend of a line

$$\begin{array}{ccc} r & s & r \\ s & t & s \\ r & s & r \end{array} \qquad \begin{array}{ccc} r & s & r \\ s & t & s \\ r & s & r \end{array} \qquad \begin{array}{ccc} r & s & r \\ s & t & s \\ r & s & r \end{array}$$

Here we require $t + 2s < 2s + 4r$ so $t < 4r$.

Second, for a diagonal line or right-angle bend in diagonal lines

$$\begin{array}{ccc} r & s & r \\ s & t & s \\ r & s & r \end{array} \qquad \begin{array}{ccc} r & s & r \\ s & t & s \\ r & s & r \end{array}$$

Here, the requirement is $t + 2r < 4s + 2r$ so $t < 4s$.

Third, acute and oblique angle corners produce, at the apex of the bend

$$\begin{array}{ccc} r & s & r \\ s & t & s \\ r & s & r \end{array} \qquad \begin{array}{ccc} r & s & r \\ s & t & s \\ r & s & r \end{array}$$

Here, the requirement is $t + s + r < 3s + 3r$ so $t < 2s + 2r$. Putting these conditions together gives

$$\max(2r,\ 2s) < t < \min(4r,\ 2r + 2s,\ 4s)$$

But $2r + 2s \geq \min(4r, 4s)$, so $\max(2r, 2s) < t < \min(4r, 4s)$ (A) is the condition for removing single pixel width lines and preserving high/low blocks of any orientation.

The smallest values of r, s, and t that satisfy (A) are $r = s = 1$; $t = 3$. This shows that a median filter, for which t is unity, and r and s are zero or unity, but r and s not both zero, cannot satisfy both conditions.

Defining the minimal filter of a subclass (that give identical output images for a given input image, for any input image) to be that filter with the smallest sum of weights, the minimal WMF that satisfies conditions 1 and 2 is

$$\begin{array}{ccc} 1 & 1 & 1 \\ 1 & 3 & 1 \\ 1 & 1 & 1 \end{array} \tag{3}$$

It should be noted that lines of width greater than one pixel may be removed by setting an appropriate spacing for the filter (3). That is, lines of width w pixels may be removed by setting a spacing of $\lceil (w + 1)/2 \rceil$ (where $\lceil x \rceil$ is the smallest integer greater than or equal to x).

In the case of Plate I, the upper satellite trail is greater than one pixel wide. Thus, the filter (3) was applied with spacing two. Plate II shows the result. The lower satellite trail has disappeared, but traces remain of the upper, wider trace. Thus, the filter was applied twice in all, giving a processed image (Plate III).

Plates IV and V show the results of applying filters (1) and (2) respectively, each twice with spacing two as for Plate III. Filter (1) leaves the satellite trials unaffected and filter (2) removes them.

The effect on the stepwedges of filters (3), (1), and (2) are shown in Plates VI, VII, and VIII respectively. These plates show the difference between the original and filtered image in each case for a magnified region around the stepwedges with intensity scaling (−10, 10) for black to white. Filters (3) and (1) changed the image very little, while Plate VIII shows that filter (2) changed the corners of the stepwedges extensively. The black areas indicate a difference of ten or more between original and filtered images.

Thus, the filters performed as predicted. That is, Plates IV and VII show that filter (1) preserves blocks and fails to remove lines. Plates V and VIII show that filter (2) removes lines but also changes parts of blocks. Plates III and VI show that filter (3) removes lines and leaves blocks unaffected.

In addition, one further benefit of filter (3) is detectable. Stars of greater extent than the filter are not completely removed by a WMF. However, the filters do affect the detected brightness, given by aperture photometry, by cutting off the narrow width peak of the transmission region. The star marked with an arrow in

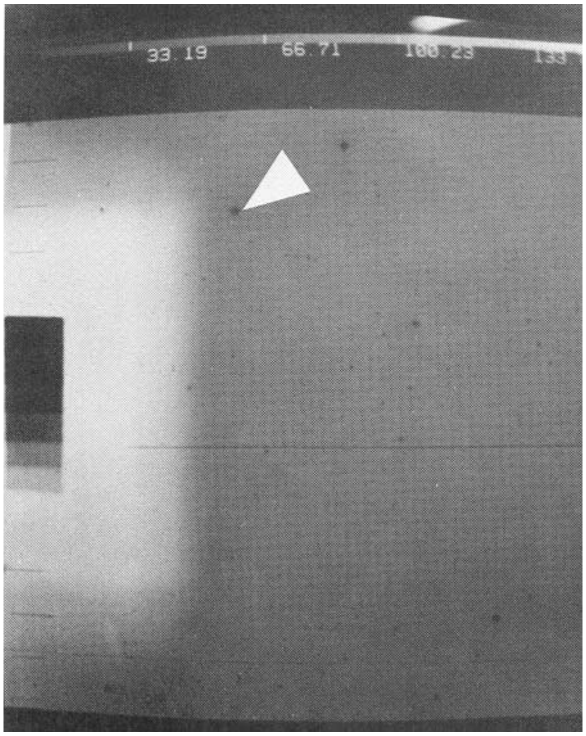

PLATE I. A 450 × 450 raster image generated by scanning part of SERC Atlas Plate 116. The arrow marks a star used for photometry in this original and in the processed images (Plates III, IV, and V). A stepwedge block is to the left of the image and two satellite trails run across its lower half. Intensity scaling is black for zero to white for 172.

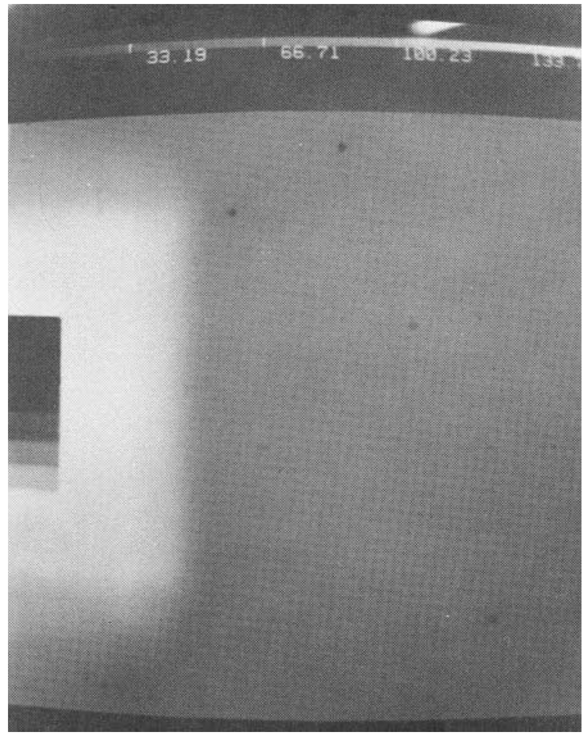

PLATE II. The image after application of filter (3), with spacing two and zero threshold.

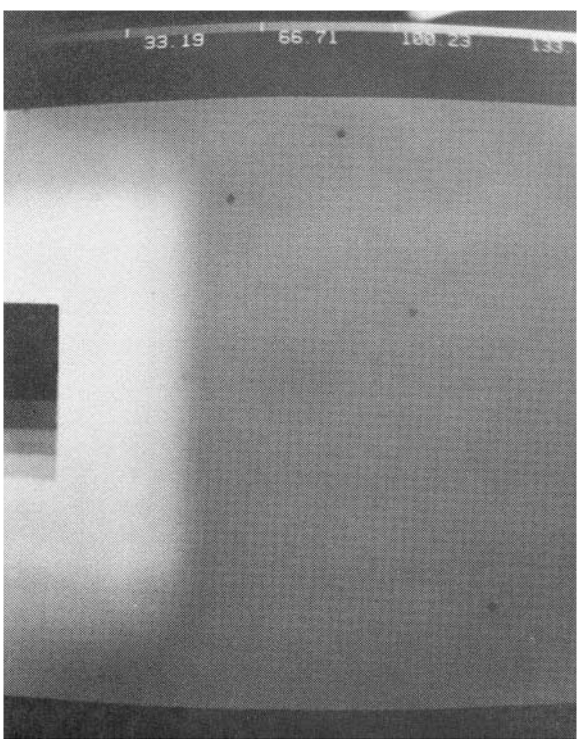

PLATE III. The image after two applications of filter (3), with spacing two and zero threshold.

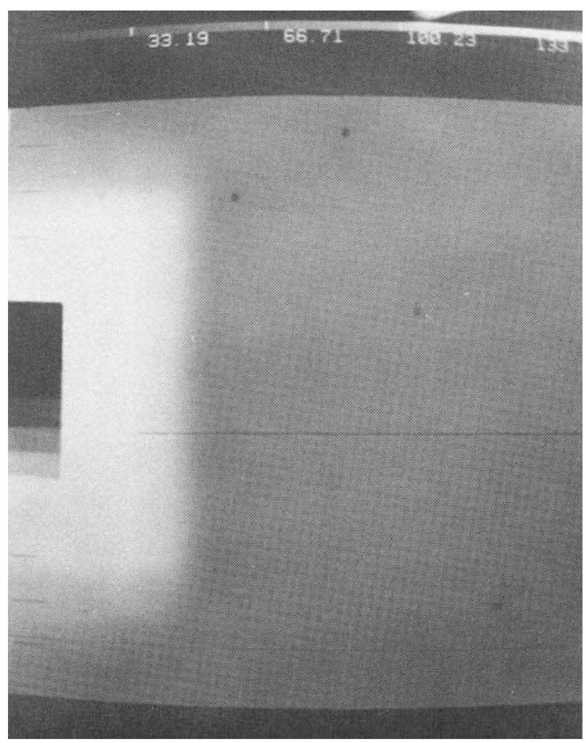

PLATE IV. The image after two applications of filter (1), with spacing two and zero threshold.

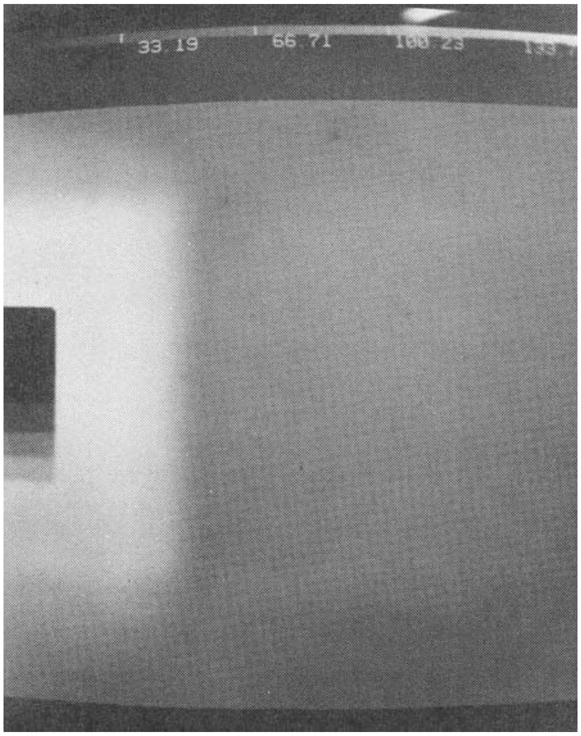

PLATE V. The image after two applications of filter (2), with spacing two and zero threshold.

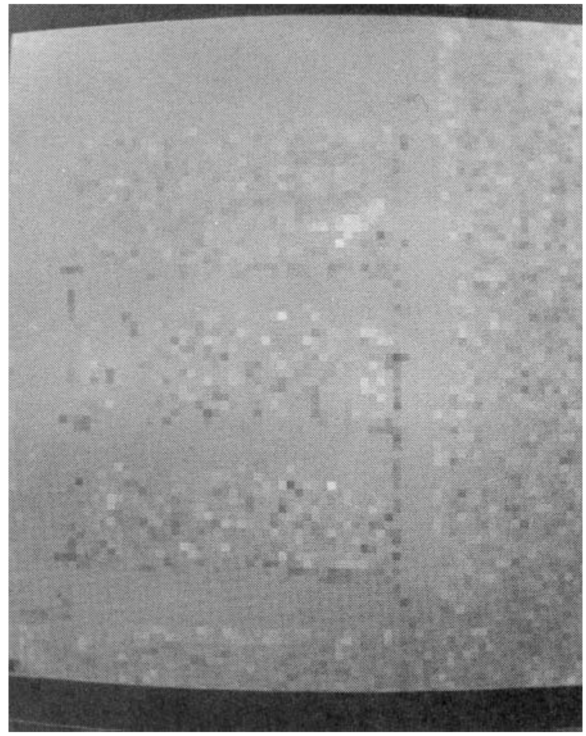

PLATE VI. The difference between the images of Plates I and III, in the region of the step wedges, with intensity scaling black for −10 or less to white for 10 or more.

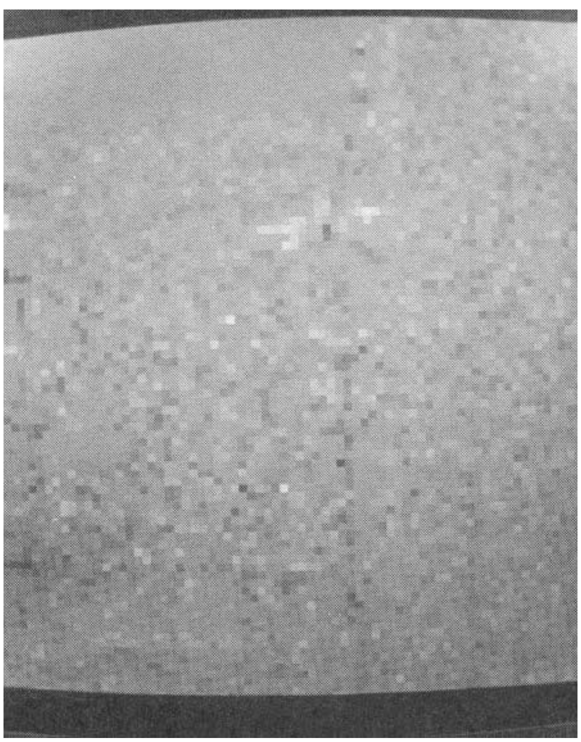

PLATE VII. The difference between the images of Plates I and IV, in the region of the step wedges, with intensity scaling as for Plate VI.

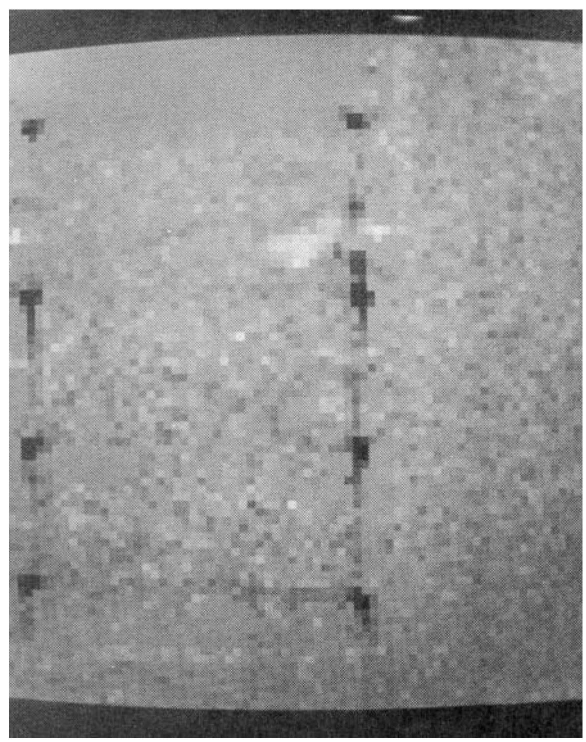

PLATE VIII. The difference between the images of Plates I and V, in the region of the step wedges, with intensity scaling as for Plate VI.

Plate I has extent greater than the applied filters. Photometry gives

Plate	Image	Apparent Magnitude
I	Original	24.632
III	Filter (3)	24.861
IV	Filter (1)	25.053
V	Filter (2)	25.546

NB one magnitude change corresponds to a factor of 2.512 in brightness.

Thus, filter (3) not only performs as originally required, but also conserves star magnitudes more effectively, allowing better photometry on the processed image. Filters (1) and (2) are not so effective in this way, filter (2) having removed over half the detected light flux from the given star.

4. ENUMERATION OF DISTINCT WMFs IN A GIVEN CLASS

Consider the WMF class of extent three, introduced in Section 3.

$$\begin{matrix} r & s & r \\ s & t & s \\ r & s & r \end{matrix}$$

At first sight, it appears that there are an arbitrarily large number of ways that such a filter can modify data as there is no limit to the number of sets of values (r, s, t), that can be assigned. On examination, this proves not to be the case.

The median value in the sorted list of $4r + 4s + t$ values can be from an r-, s- or t-occurring value. Let there be $m1$ r-occurring, $m2$ s-occurring, and $m3$ t-occurring values to the left of the value that occurs as the median.

Thus, $0 \le m1, m2 \le 4$ and $0 \le m3 \le 1$, in general. And if the median is r-occurring, $m1 \le 3$; s-occurring, $m2 \le 3$; and t-occurring, $m3 = 0$.

If $(m1, m2, m3, x)$ is denoted by a sorted list such that there are $m1$ r-, $m2$ s- and $m3$ t-occurring values to the left of the x- occurring value containing the median, then, $m1 \cdot r + m2 \cdot s + m3 \cdot t < (4 - m1) \cdot r + (4 - m2) \cdot s + (1 - m3) \cdot t$ and $m1 \cdot r + m2 \cdot s + m3 \cdot t + x > (4 - m1) \cdot r + (4 - m2) \cdot s + (1 - m3) \cdot t - x$; that is, $(2 \cdot m3 - 1) \cdot t + (2 \cdot m2 - 4) \cdot s + (2 \cdot m1 - 4) \cdot r < 0$ and $(2 \cdot m3 - 1) \cdot t + (2 \cdot m2 - 4) \cdot s + (2 \cdot m1 - 4) \cdot r + 2 \cdot x > 0$ (x may be r, s or t, giving three inequalities with ">" and one with "<").

Given $(m1, m2, m3)$, limits then result on r, s, and t. Similarly, given a filter, (r, s, t), then only certain combinations of $(m1, m2, m3, x)$ are possible. The multipliers for r, s, and t are the terms in brackets of the form $(2 \cdot mi - c)$. The multiplier for

(i) t has a value from the set $\{-1, 1\}$
(ii) s has a value from the set $\{-4, -2, 0, 2, 4\}$
(iii) r has a value from the set $\{-4, -2, 0, 2, 4\}$

The inequalities are simplified by dividing through by $(2 \cdot m3 - 1)$. All possible inequalities are then $(t + n1 \cdot r + n2 \cdot s < 0, t + n1 \cdot r + n2 \cdot s > 0$: $n1, n2$ in $\{-4, -2, 0, 2, 4\})$. They are listed in Table I. Combinations of these inequalities express possible orders for the values and which value contains the median.

Inequalities marked $*$ cannot be satisfied and inequalities marked # are trivially satisfied because $t > 0$ and $r, s \ge 0$.

Now consider a straight line, $L(x, y; r, s, t) = t + r \cdot x + s \cdot y = 0$, where r, s, and t are WMF filter weights.

Then L has nonpositive slope in the (x, y) plane (as $r, s > 0$) and will partition the points $(P = (n1, n2)$: $n1, n2$ in $\{-4, -2, 0, 2, 4\})$.

L will not pass through any of the Ps as $\{(n1, n2)\}$ are such that $L(n1, n2; r, s, t) \ne 0$, as shown above.

Thus, the Ps below and to the left of L represent inequalities $t + n1 \cdot r + n2 \cdot s < 0$, so $L(n1, n2; r, s, t) < 0$. Also, the Ps above and to the right of L represent inequalities $t + n1 \cdot r + n2 \cdot s > 0$, so $L(n1, n2; r, s, t) > 0$. Figure 1 shows the arrangement of points and a typical L.

Now L cannot pass to the right and above any point, Px, marked with an x in Figure 1 as that implies $L(Px) < 0$, which is not possible due to the restrictions on r, s, and t. For instance, L to the right and above the origin implies that $L(0, 0; r, s, t) < 0$. That is, $t + 0 \cdot r + 0 \cdot s < 0$, so $t < 0$; but $t > 0$.

All lines L, defined by (r, s, t), that partition the Ps in the same way correspond to the same set of inequalities from Table I being satisfied, that is, they will give the same median from some given set of data values.

Thus, all sets of values (r, s, t) that result in $L(x, y; r, s, t) = 0$ dividing the Ps in the same way correspond to filters that act in the same way on given data. In other

TABLE I. Inequalities for *rst* Filters

$t - 4r - 4s < 0$	$t - 4r - 4s > 0$
$t - 2r - 4s < 0$	$t - 2r - 4s > 0$
$t \quad - 4s < 0$	$t \quad - 4s > 0$
$t + 2r - 4s < 0$	$t + 2r - 4s > 0$
$t + 4r - 4s < 0$	$t + 4r - 4s > 0$
$t - 4r - 2s < 0$	$t - 4r - 2s > 0$
$t - 2r - 2s < 0$	$t - 2r - 2s > 0$
$t \quad - 2s < 0$	$t \quad - 2s > 0$
$t + 2r - 2s < 0$	$t + 2r - 2s > 0$
$t + 4r - 2s < 0$	$t + 4r - 2s > 0$
$t - 4r < 0$	$t - 4r > 0$
$t - 2r < 0$	$t - 2r > 0$
$t < 0$ *	$t > 0$ #
$t + 2r < 0$ *	$t + 2r > 0$ #
$t + 4r < 0$ *	$t + 4r > 0$ #
$t - 4r + 2s < 0$	$t - 4r + 2s > 0$
$t - 2r + 2s < 0$	$t - 2r + 2s > 0$
$t \quad + 2s < 0$ *	$t \quad + 2s > 0$ #
$t + 2r + 2s < 0$ *	$t + 2r + 2s > 0$ #
$t + 4r + 2s < 0$ *	$t + 4r + 2s > 0$ #
$t - 4r + 4s < 0$	$t - 4r + 4s > 0$
$t - 2r + 4s < 0$	$t - 2r + 4s > 0$
$t \quad + 4s < 0$ *	$t \quad + 4s > 0$ #
$t + 2r + 4s < 0$ *	$t + 2r + 4s > 0$ #
$t + 4r + 4s < 0$ *	$t + 4r + 4s > 0$ #

Inequalities arising from all possible positions of the median as an r-, s- or t-occurring value in the ordered list of $4r + 4s + t$ values. Inequalities marked $*$ cannot be satisfied and those marked # are trivially satisfied.

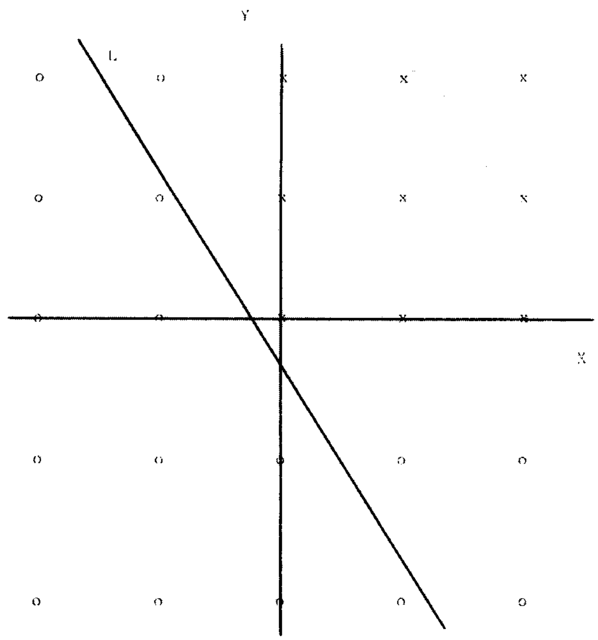

FIGURE 1. The Points P of Section 4 and a Typical Line L Passing between Them

words, given ($r1$, $s1$, $t1$) and ($r2$, $s2$, $t2$) such that $L(x, y; r1, s1, t1) = 0$ and $L(x, y; r2, s2, t2) = 0$ divide the Ps in the same way, then the filters

$$\begin{array}{ccc} r1 & s1 & r1 \\ s1 & t1 & s1 \\ r1 & s1 & r1 \end{array} \quad \text{and} \quad \begin{array}{ccc} r2 & s2 & r2 \\ s2 & t2 & s2 \\ r2 & s2 & r2 \end{array}$$

will produce identical results from the same input data.

Hence, the number of distinctly acting WMFs of the form

$$\begin{array}{ccc} \mathrm{r} & \mathrm{s} & \mathrm{r} \\ \mathrm{s} & \mathrm{t} & \mathrm{s} \\ \mathrm{r} & \mathrm{s} & \mathrm{r} \end{array}$$

is equal to the number of ways of partitioning the Ps with a straight line of nonnegative slope that passes below and to the left of the origin in the (x, y) plane.

A FORTRAN77 program has been written to enumerate the partitions of which there are 53. These are shown in Figure 2. In Table II are listed representative members of each of the 53 types.

Note that the list includes the median filters already introduced, the median filter that uses four diagonal neighbors (1, 0, 1), and also the filter that has no effect on the data, (1, 1, 9), for which $t - 4 \cdot r - 4 \cdot s > 0$. That is, $t > 4 \cdot r + 4 \cdot s$ and so the t-occurring value always provides the median. As, in this case, the t-occurring value is the original data value, the data is unchanged.

Another generalization can be made. The number of distinct filters given is for $4r$-, $4s$- and $1t$-occurring values with a given pattern. That is, there are 53 distinct WMFs of the form

$$\begin{array}{ccc} \mathrm{r} & \mathrm{s} & \mathrm{r} \\ \mathrm{s} & \mathrm{t} & \mathrm{s} \\ \mathrm{r} & \mathrm{s} & \mathrm{r} \end{array}$$

As the argument concerning partitions did not depend on the layout of the skeleton, but only on the number of r, s, and t values, it is also true to state that:

Given any WMF such that $a(i, j) = r$ for 4 values,
$a(i, j) = s$ for 4 values,
$a(i, j) = t$ for 1 value,
r, s nonnegative integers,
t positive odd integer, and
$a(i, j) = 0$ otherwise,

then there are exactly 53 ways that such a filter can act, according to the values of r, s, and t. For example, the WMF defined by the array

$$\begin{array}{ccccc} \mathrm{r} & \mathrm{r} & \mathrm{r} & \mathrm{r} & 0 \\ \mathrm{t} & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathrm{s} & 0 & \mathrm{s} \\ 0 & 0 & 0 & \mathrm{s} & 0 \\ \mathrm{s} & 0 & 0 & 0 & 0 \end{array}$$

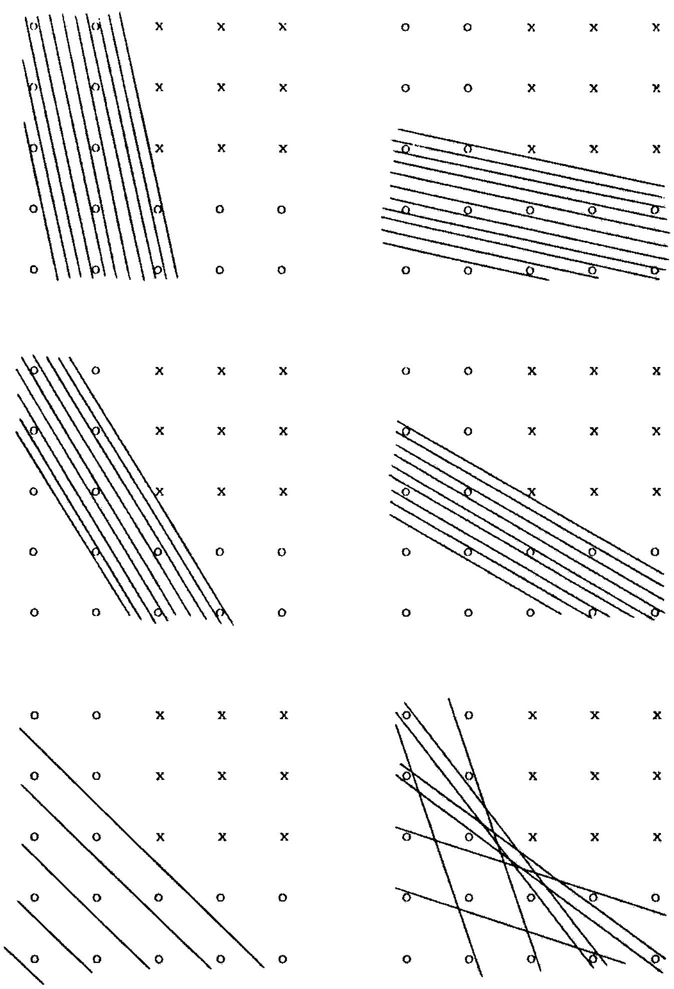

FIGURE 2. The 53 Partitions of the Points P of Section 4

TABLE II. Distinct *rst* Filters

r	*s*	*t*	*r*	*s*	*t*	*r*	*s*	*t*	*r*	*s*	*t*
1	1	9	4	1	11	1	1	3	2	3	3
1	1	7	5	2	15	1	3	5	3	4	3
1	0	3	2	1	7	2	3	5	1	1	1
2	1	9	1	0	1	1	2	3	4	3	5
4	1	13	4	1	5	1	3	3	4	3	3
3	1	11	5	2	11	1	4	3	3	2	3
1	2	9	3	1	5	1	4	5	1	3	1
1	3	11	2	1	5	3	2	5	2	3	1
1	4	13	3	2	9	0	1	1	3	1	1
0	1	3	1	2	5	2	1	3	3	2	1
1	1	5	2	5	11	3	1	3	4	2	3
1	2	7	2	3	9	4	1	3			
2	5	15	2	3	7	2	4	3			
1	4	11	3	2	7	3	4	5			

Examples of each of the 53 distinctly acting filters with four *r*-, four *s*-, and one *t*-occurring values.

can act in exactly 53 ways on given data depending on the values of *r*, *s*, and *t*.

The use of a filter skeleton with only three independent parameters and many zero weights is not frequently useful, but has applications in at least two ways: First, in the removal of lines or objects of width greater than one pixel, as described in Section 2. The idea of spacing allowed the removal of redundant zeros for the example given, but filters of the form

$$\begin{array}{ccccc} 0 & 0 & r & 0 & 0 \\ 0 & 0 & s & 0 & 0 \\ r & s & t & s & r \\ 0 & 0 & s & 0 & 0 \\ 0 & 0 & r & 0 & 0 \end{array}$$

may be of use for some purposes. Second, special filters may be used for edge or corner values, such as

$$\begin{array}{ccccc} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & t & s & r \\ 0 & 0 & s & s & r \\ 0 & 0 & r & r & s \end{array}$$

An important special case, of which the filter (3) of Section 3 is an example, is when $r = s$, so the WMF is written as (r, r, t), or more explicitly as

$$\begin{array}{ccc} r & r & r \\ r & t & r \\ r & r & r \end{array}$$

Here, the slope of L is -1, as $r = s$, and there are only five distinct ways of partitioning the Ps, as is shown in Figure 3.

Hence, there are only five distinctly acting WMFs of the form

$$\begin{array}{ccc} r & r & r \\ r & t & r \\ r & r & r \end{array}$$

Their minimal members are $(r, s, t) = (1, 1, 1)$, $(1, 1, 3)$, $(1, 1, 5)$, $(1, 1, 7)$, and $(0, 0, 1)$. Note that $(1, 1, 9)$, equivalent to $(0, 0, 1)$, is not minimal.

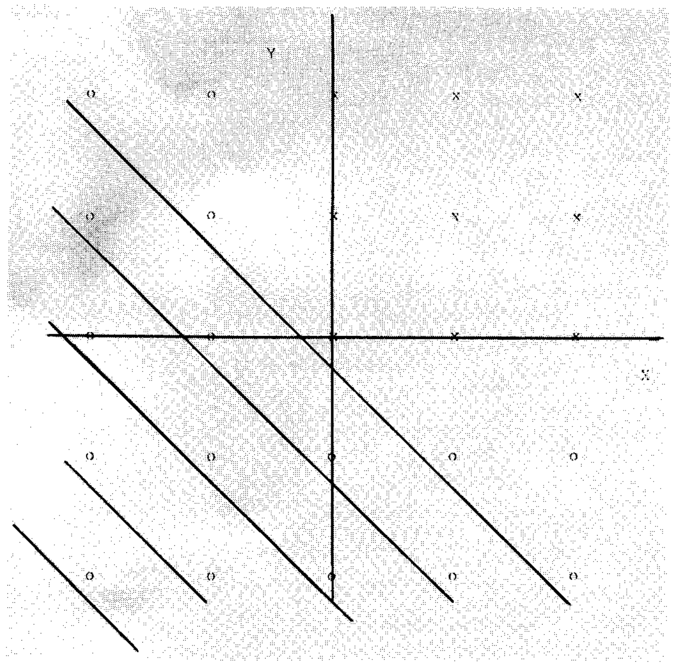

FIGURE 3. The Five Ways of Partitioning the Points *P* of Section 4 when $r = s$

5. RESULTS OF USING WMFs AND FURTHER DESIRABLE PROPERTIES

Using a pseudorandom number generator, test data were generated. The test data were a 25×25 array of integers, randomly distributed between 90 and 110. Each filter in Table II was iteratively applied to the data until no change was made on successive iterations with zero threshold, up to a maximum of 20 iterations. If convergence had not taken place after 20 iterations, the number of values changed between steps 19 and 20, (# vals.), was counted. Sometimes, further convergence could take place, but usually an oscillatory state had been reached, the problem of which is discussed in Section 6.

A threshold of zero was used. That is, a value was replaced by the median, however small their absolute difference. The spacing used was unity. Also calculated are the parameters SIDESQ and DIAGSQ, which are defined for the example data as

$$\begin{aligned} \text{SIDESQ} = \sum_{i=2}^{24} \sum_{j=2}^{24} \{&(d(i, j) - d(i-1, j))^2 + (d(i, j) \\ &- d(i+1, j))^2 + (d(i, j) - d(i, j-1))^2 \\ &+ (d(i, j) - d(i, j+1))^2\} \end{aligned}$$

$$\begin{aligned} \text{DIAGSQ} = \sum_{i=2}^{24} \sum_{j=2}^{24} \{&(d(i, j) - d(i-1, j-1))^2 + (d(i, j) \\ &- d(i-1, j+1))^2 \\ &+ (d(i, j) - d(i+1, j+1))^2 \\ &+ (d(i, j) - d(i+1, j-1))^2\} \end{aligned}$$

The expectation for a uniform background with uniformly distributed noise is that

$$E(\text{SIDESQ}) = E(\text{DIAGSQ})$$

TABLE III. Action of Each of the Filters of Table II When Applied to the Same Test Data

r	s	t	Iterations	# vals.	SIDESQ	DIAGSQ	R
1	1	9	1		152382	150095	0.03025
1	1	7	2		122753	122193	0.00914
1	0	3	3		99946	69379	0.74642
2	1	9	5		98668	85903	0.27797
4	1	13	2		97818	69915	0.68435
3	1	11	2		94862	73454	0.51712
1	2	9	5		88571	101703	−0.27739
1	3	11	5		77818	99180	−0.48990
1	4	13	3		76526	99774	−0.53680
0	1	3	3		76430	100554	−0.55555
1	1	5	10		57341	60551	−0.10899
1	2	7	8		44524	54179	−0.39505
2	5	15	6		43719	54421	−0.44144
1	4	11	6		43439	54469	−0.45642
4	1	11	7		39328	32595	0.37777
5	2	15	7		38808	32719	0.34300
2	1	7	11		36984	35144	0.10211
1	0	1	*	14	35136	7199	4.67579
4	1	5	7		24256	8196	2.62160
5	2	11	12		17314	15258	0.25350
3	1	5	9		16889	8584	1.45924
2	1	5	12		16631	15218	0.17781
3	2	9	13		15216	15406	−0.02482
1	2	5	11		10077	14460	−0.73806
2	5	11	11		10037	14440	−0.74359
2	3	9	13		9979	13916	−0.67744
2	3	7	8		9835	13908	−0.70699
3	2	7	11		9676	13082	−0.61236
1	1	3	9		9601	13456	−0.68801
1	3	5	9		9269	15144	−1.02178
2	3	5	5		9219	14962	−1.00679
1	2	3	5		9195	14906	−1.00423
1	3	3	*	12	7805	12754	−1.02212
1	4	3	*	21	7570	12454	−1.03734
1	4	5	*	8	7548	12368	−1.02830
3	2	5	8		7518	8033	−0.13261
0	1	1	*	28	7471	12242	−1.02833
2	1	3	8		5617	6375	−0.25385
3	1	3	8		3261	3600	−0.19812
4	1	3	*	40	1610	1701	−0.11002
2	4	3	*	6	1093	1610	−0.79413
3	4	5	20	(a)	1042	1500	−0.74487
2	3	3	20	(b)	1042	1500	−0.74487
3	4	3	20		1041	1534	−0.79497
1	1	1	20		1040	1540	−0.80544
4	3	5	20		1034	1544	−0.82354
4	3	3	20		1031	1534	−0.81578
3	2	3	20		1025	1534	−0.82840
1	3	1	*	10	938	1420	−0.85330
2	3	1	*	4	917	1350	−0.79293
3	1	1	*	72	819	1104	−0.60614
3	2	1	*	9	723	1030	−0.72268
4	2	3	*	10	719	1054	−0.78376

(a) gives 17798, 25434 after one iteration, $R = -0.72927$.
(b) gives 17922, 25618 after one iteration, $R = -0.72983$.

For the test data, a desirable property of the filter is that the quantity $R = (\text{SIDESQ}\cdot\text{SIDESQ} - \text{DIAGSQ}\cdot\text{DIAGSQ})/(\text{SIDESQ}\cdot\text{DIAGSQ})$ remain close to the initial value. This is the R value for the (1, 1, 9) filter which does not change the data. Preservation of R at approximately zero is desirable because it indicates that the filter is not changing the relative variation in side and diagonal neighbor values. As an example, consider an image of uniform waves of the form

```
      .    .    .    .
· · 101  102  101  102 · ·
· · 101  102  101  102 · ·
· · 101  102  101  102 · ·
           .    .    .
```

That is, (101, 102) alternate across the image. This is a highly structured image made up of short wavelength variations of constant amplitude. This ordering is indicated by -1.5 for the value for R.

For the test data, results are shown in Table III. It can be seen that R varies widely according to which filter is used, though some filters provide the desirable property of R little changed. Note that the popular median filters, (0, 1, 1) and (1, 1, 1), filters (1) and (2), perform badly in this regard. Also, (0, 1, 1) has not converged and (1, 1, 1) has taken a full 20 iterations to converge. On another test data set, generated in the same way and using a different seed for the pseudorandom number generator, neither of the filters, (1) and (2), converged.

The results are listed in order of decreasing size of the value of SIDESQ. In particular, the filter (1, 1, 9), that does not change the data, only processes one iteration and the values of SIDESQ and DIAGSQ are those for the original data set.

Various boundary strategies, in which the filter was not used on boundary values, or in which a wide strip of boundary values was not used in calculating SIDESQ and DIAGSQ, have been tried, but the results in terms of the values for R are not qualitatively different.

It is also found that where R is markedly different from zero, the difference is cumulative over several iterations and, for most filters, the first iteration does not change R from zero to a large degree. This is of relevance in the use of mixed filter strategies, discussed in Section 6.

The wide variation in the final values of SIDESQ and DIAGSQ should be noted. This arises because the likelihood of replacement of the value by the median changing the data value depends, approximately, on the ratio of $(4r + 4s)$ to t. The dependency is only approximate as there are a limited number of distinct filters in the class. The final values of SIDESQ and DIAGSQ give a measure of the effectiveness of the filter in achieving backgrounding. In the example data set, the background is flat and the values should be close to zero.

6. FURTHER CONSIDERATIONS

6.1 Mixed Strategies

Conditions 1 and 2, given in Section 3, determine a range of WMFs, according to the inequality $\max(2r, 2s) < t < \min(4r, 4s)$. However, such filters do not generally satisfy the requirement, considered in Section 5, of keeping R near zero, when used iteratively for backgrounding as can be seen from Table III.

An answer is to use one filter, such as (1, 1, 3), for one iteration, to clean lines while preserving block corners. This changes R only slightly in one pass over the data. Backgrounding may then continue iteratively with a filter that preserves blocks and has good R performance. The block preservation condition is

$$t > \max(2r, 2s)$$

The filter (3, 2, 9) satisfies this condition and has good R performance, while reducing SIDESQ and DIAGSQ to about 10 percent of their original values.

Also, the occurrence of four high (low) values spaced in a square of size equal to the filter spacing will cause the filter to interpret the values as belonging to a block. The WMF should, therefore, be used over a variety of spacings on the same image to allow for this effect.

With the filter used as specified in Section 2, there is no significant cost in computation of varying the filter skeleton or spacing, provided all rows needed for applying the filter at a given point can be held in main memory simultaneously. Changing the filter used at each iteration is also of negligible cost in time and space, as the required filter at iteration i may be stored in (and read from) the ith level of an array of filter coefficients.

Further, to gain computational efficiency for a general $(2n + 1) \times (2n + 1)$ filter skeleton with a sum of weights S, as in Section 2, only $(2n + 1) \cdot (2n + 1)$ values have to be sorted and not S. The reason is that only the distinct values need be sorted, and a copy of the number of occurrences kept with each value.

6.2 Oscillatory Values

In Section 5, many filters had not converged after 20 iterations. In most cases, this was due to sets of values oscillating in position on successive iterations. This may happen in a number of ways. A simple example is

2	2	1	1	1
2	2	2	1	1
1	1	2	1	1
1	1	2	2	2
1	1	1	2	2

Using the filter (1, 1, 1) on the internal values produces

2	2	1	1	1
2	2	1*	1	1
1	2*	2	2*	1
1	1	1*	2	2
1	1	1	2	2

The * values are those that have changed. Repeating the use of the WMF returns the data to its original state. An oscillatory state has been entered.

Filters with a low $t/(4r + 4s)$ value are more likely than others to show this effect. It arises when a surfeit of high values in the neighborhood of a low point exists, causing it to be changed to a high value, but enough of its high neighbors have been changed to low values to result in a change back to low on the next iteration.

For (1, 1, 1) on a low (L) point, at least five neighbors must be high (H) to change it. Once changed to H, at least five of its neighbors must be L to change it back.

That is, at least two neighbors must have been changed from H to L.

For a filter such as (1, 1, 3), there are a total of 11 values. If a point is L, six neighbors must be H to change it to H. Then to change it back to L, at least six neighbors must be L, so no more than two can be H. Thus, four neighbors must change from H to L and back on each iteration, which can only occur indefinitely for data of infinite extent, because of boundary effects. Thus, (1, 1, 3) and other filters with a higher value of t in comparison to $4r + 4s$ will always converge.

For filters other than the above type, the same general arguments apply. That is, the relative size of the central value, $a(0, 0)$, which gives the number of copies of the current value taken, in comparison with the sum of other filter weights, will be related to the likelihood for the filter of avoiding oscillation. An extreme example is the filter

$$\begin{matrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{matrix}$$

which leaves the data unchanged and can never oscillate. The central value divided by the sum of the others is undefined (or "infinite"). At the other extreme, a filter

$$\begin{matrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{matrix}$$

will generally not converge, as the central value is zero, and so a data value is always replaced by a neighbor. However, some particular image data sets will give an unchanged image under the filter. An example is a uniform image where a data value is either unchanged or replaced by an equal value for any WMF.

7. CONCLUSIONS

Weighted median filters (WMFs) provide a way of backgrounding a raster image or removing unwanted content while preserving specified desirable features of the image. That is, a WMF can be designed to work in a controlled way. A WMF can be designed to perform a specific task or a range of tasks, depending on the number of free parameters available. The parameters of a WMF are the filter skeleton weights which are specified by the restrictions of the desired application. For any WMF skeleton, there are a fixed number of ways that the WMF can act on the data, and all filters of that form act in one of numerable ways.

Acknowledgments. The author would like to acknowledge assistance from the Royal Observatory, Edinburgh where this work was begun, and to the staff of the Image Data Processing Unit of the Observatory for providing image data and analysis facilities.

REFERENCES

1. Tukey, J.W. *Exploratory Data Analysis*. Addison-Wesley, Reading, Mass., 1974.
2. Pratt, W.K. *Digital Image Processing*. John Wiley and Sons, Inc., New York, 1978.
3. Stobie, R.S. Royal Observatory Edinburgh, internal publication, 1981.
4. Martin, R., and Lutz, R.K. Background following on Schmidt plates using a digital filtering technique. International Workshop on Image Processing in Astronomy, Proceedings 5th. Coll. on Astrophysics, Trieste, 1979.

CR Categories and Subject Descriptors: I.4.3 [**Image Processing**]: Enhancement—*filtering*

General Terms: Design

Additional Key Words and Phrases: median filters, weighted median filters, feature removal and retention, backgrounding

Received 11/82; revised 1/83; accepted 1/84

Author's Present Address: D.R.K. Brownrigg, Dept. of Mathematics, University of London Goldsmiths' College, London SE14 6NW, U.K.