OPTIMUM NON LINEAR DOCUMENT RESTORATION THROUGH LINEAR GREYSCALE OPERATORS

Stephen Marshall
Dept of Electronic and Electrical Engineering
University of Strathclyde, Glasgow, UK
Tel: +44 141 548 2199; Fax: +44 870 163 9219
e-mail: s.marshall@eee.strath.ac.uk

ABSTRACT

Non linear image processing operators give excellent results in a number of image processing tasks such as restoration and object recognition. However they are frequently excluded from use in solutions because the system designer does not wish to introduce additional hardware or algorithms and because their design can appear to be ad hoc. This paper explains how various non linear image processing operators may be implemented on a basic linear image processing system using only convolution and thresholding operations.

As well as describing novel algorithms for implementation within a linear system the paper also explains how the optimum filter parameters may be estimated for a given image processing task.

1 INTRODUCTION

Mathematical morphology [1] consists of a powerful set of tools for image processing which may be used for many tasks including noise reduction and object recognition. However its definition in terms of set theory and subsequently in terms of lattices can make it appear remote from more mainstream operations such as linear filtering. Other non linear methods such as order statistic and weighted order statistic (WOS) [2] filters are excellent at removing noise and preserving image structure but only the special case of the median appears to be in widespread use. The sorting operations are thought to be computationally expensive and the hardware implementation is perceived as comparator based and inherently incompatible with linear multiply-accumulate architectures.

Frequently in assembling a large hardware or software solution to an image processing problem the system designer chooses to discount non linear operations as they require additional hardware or software extensions. This paper demonstrates strategies for decomposing a number of non linear operators so that they may be implemented through standard linear hardware and software configurations. In particular mathematical morphology, rank order statistic filters including the median and weighted median operators will be discussed.

BINARY IMAGE PROCESSING

Many documents including pages of text and faxed items may be adequately represented by only two intensity levels. The non linear image processing techniques mentioned above may be used for a number of common tasks including restoration of degraded text, noise removal, OCR and object recognition.

ORDER STATISTIC FILTERS

All analysis in this paper will be based on sampled values on a 2D Euclidean grid. It involves filtering an image *I(m,n)* with *M´N* pixels, where *0 £m£ M-1* and *0 £n£ N-1*.

In all cases the image *I(m,n)* is processed by a filter defined within an overlapping sliding window *B(k,l) where -K £ k £ K , -L £ l £ L*. The filter window therefore consists of a region of support of *(2K+1) ´ (2L+1)* pixels with origin at *B(0,0)*. The region of support of *B(k,l)* has been assumed to be rectangular with odd dimensions for computational simplicity. Where a non rectangular window is required, i.e. a cross or circle, the pixels within the overall region of support of the window which are not to be used in the calculation are set to **0** otherwise they are set to **1**. The order (or size) of the filtering window is expressed as $|B|$ and is calculated as the sum of elements in *B*. For the basic rank order filter y_r applied to a set of input values $\boldsymbol{x}=(x_0, x_2,\ldots, x_{|B|-1})$ the output is defined as the *r* th largest value in the input window.

Well known special cases of rank order filters are the minimum when *r=1*, the maximum when *r* = $|B|$ and the median when *r* = $(|B|+1)/2$. For simplicity it is assumed that $|B|$ always remains odd. More complex examples of rank order filters can be formed by (a) duplicating the input variables which results in weighted order statistic (WOS) filters [2] and (b) forming functions of the ordered variables which results in filters such as the Huber [3] and ∝ trimmed mean [4]. Further details of these filters can be found in the references.

Where the filter is applied to binary image pixels the set of $\boldsymbol{x}$ is, x_i *Î{0,1}* for all *i*. Therefore the input window *B(k,l)* will contain $|B|$ variables which can only take the values **0** and **1**. Assuming that there are *z*, **0**s and $|B|$-*z*, **1**s the set of inputs from the window after rank ordering is *{1, 1,……1, 0, 0……0}* i.e. a string of **1**s followed by a string of **0**s. The filter output is equal to the *r* th rank order pixel in the window. This will therefore be **1** if the number of **1**s in the window *B* is greater than or equal to *r* i.e. if $|B|$-*z* ³ *r* otherwise the filter output will be **0**.

The rank order filter y_r applied to the image I, within a window, *B(k,l)* centred around the reference pixel, *B(0,0)* can be written as,

$$ø_r(I) = \begin{cases} 1 \quad if \quad \sum_{(k,l)\in B(k,l)=1} I(m+k,n+l) \ \geq \ r \\ 0 \qquad\qquad\qquad\qquad otherwise \end{cases} \qquad (1)$$

Therefore in the binary case the rank order filter y_r reduces from a sorting process to a count of the pixels of *I* falling within the window *B*, followed by a threshold.

IMPLEMENTATION OF MORPHOLOGICAL AND RANK ORDER OPERATORS

Consider the following statement:

All rank order filters including the median and a number of the fundamental SSP (set-set processing) tasks within mathematical morphology may be implemented simultaneously for binary images via a single linear (multiply-accumulate) operation carried out between the original image, I, and the filter window, B, followed by thresholding at an appropriate level.

The convolution operator is central to all linear software and hardware image-processing systems. The convolution $H(m,n)$ between an image I and window B is defined as follows:

$$H(m,n) = I * B = \sum_{k=-K}^{K} \sum_{l=-L}^{L} B(k,l)\, I(m+k, n+l) \qquad (2)$$

The above operator is used in edge detection, linear smoothing and sharpening.

In order to implement the morphological operators and rank order filters, the image I and the filter window, B, are convolved to produce a single image, H. Although both I and B are binary, the result of their convolution, H, is a greyscale image with pixel values in the range, 0 to $|B|$.

$$H(m,n) = I * B = \sum_{(k,l)\in B(k,l)=1} I(m+k, n+l) \qquad (3)$$

The image, H, will be shown to consist of a stack of all the outputs of the rank order filter y_r , for every value of r. The required rank order filtered output image, $y_r(I)$, may be obtained by thresholding the image, H, at the appropriate value of r.

The correlation of the image, I, with the window, B, is equivalent, in the binary case, to an operation which counts the number of pixels within the window, B, for which $I=1$, and sets the corresponding pixel in image H, to this value. This leads to the greylevel image H, in which the pixel values reflect the extent of window occupancy in the original image I.

The binary images resulting from filtering by y_r, i.e. the rank order, morphological and median filters are obtained by thresholding H at the appropriate level, r. The following filters may be obtained,

Rank order filter	$y_r = T^{\,r}\,[\,I*B]$	(4)
Median Filter	$Med\,B\,(I) = T^{\,(\lvert B\rvert+1)/2}\,[\,I*B]$	(5)
Dilation	I Å $B = T^{\,1}\,[\,I*B]$	(6)
Erosion	$I \ominus B = T^{\,\lvert B\rvert}\,[\,I*B]$	(7)

where $T^{\,t}[N]$ is a thresholding function defined as

$$T^{\,t}[N] = \begin{cases} 1 & if \quad N \geq t \\ 0 & Otherwise \end{cases}$$

An example is shown in Figure 1. Figure 1(a) shows a simple original image and Figure 1(b) shows a noise corrupted version. The objective is to filter the noisy image in order to recover the original or an image which is as close as possible to it. As can be seen in Figure 1(c) the resulting greyscale image contains, at each level, every rank order filter including the erosion, the dilation and the median. In this simple case the median is the best filter as it recovers the original image precisely.

WEIGHTED MEDIAN FILTER

In order to preserve finer image structure such as corners and straight lines it is often necessary to give the pixels at some locations within the window a greater weighting.

A modification of the median filter is the weighted median (or centre weighted median) [5], in which the pixel derived from the central location of the window is included in the pixel list an increased number of times compared to the other pixels. Consider the case of a 3×3 weighted median filter with weighting W and window locations labelled x_0 to x_8 such that the centre value is x_4. For binary images the weighted median filter may be expressed as,

$$wmed(\mathbf{x}) = \begin{cases} 1 \quad if \quad W x_4 + \sum_{i\neq 4} x_i \geq \dfrac{(W+9)}{2} \\ 0 \qquad\qquad\qquad\qquad otherwise \end{cases} \qquad (8)$$

$$\text{where } \sum_{i\neq 4} x_i = \sum_{i=0}^{i=3} x_i + \sum_{i=5}^{i=8} x_i$$

In the same way as the standard median filter was applied to binary images through linear convolution and a threshold function, the weighted median filter can also be implemented in this way. The only modification required is to set the window values to the appropriate weightings so in the case above $B(0,0) = W$ and $B(k,l)=1$ for all *k¹0* and *l ¹0*. The filter output can then be written as,

$$wmed\ B_w\,(I) = T^{\,(\lvert Bw\rvert+1)/2}\,[\,I*B_w] \qquad (9)$$

where B_w is the filter window including the weighted values and $|\,B_w|$ is the sum of all the values in the window.

So although the weighting, W, applied in the weighted median filter refers to the number of times the centre pixel is repeated, in the binary case, the same output may be achieved by using, W, as a multiplicative weighting of the centre pixel. Therefore in the binary case, not only does the sorting operation simplify to a basic count, but also the repetition operator is replaced by a multiplicative weighting. For a weighted median *3x3* filter it can be shown [6] that the W must be in the range *1£ W £ 7*.

OPTIMUM RANK ORDER FILTERS

Suppose a filter is defined based on the Hamming weight of the vector, $|\mathbf{x}|$. Then filters are of the form $f(x)$, and there are only $n+1$ possible inputs for which it is required to determine the filter value. These filters will be called weight filters, and the optimal weight filter is given by

$$f_{opt}(\mathbf{x}) = \begin{cases} 1, & \text{if } P(Y=1\,|\,|\,\mathbf{x}\,|) \geq 0.5 \\ 0, & \text{if } P(Y=1\,|\,|\,\mathbf{x}\,|) < 0.5 \end{cases}$$

The filter will be correct for at least 50% of the inputs. The MAE of the optimum weight filter is summed over the cases where it gives the incorrect output:

$$MAE < I_0, f_{opt}(I) >= \sum_{\mathbf{x}} P(\mathbf{x}) \min\langle P(y=0 \mid |\mathbf{x}|), P(y=1 \mid |\mathbf{x}|)\rangle \quad (10)$$

The weight filter, $f_{opt}(\boldsymbol{x})$, is sub optimal compared to the optimal of all filters defined in the window n. This is because it has been constrained to consider only the weight of the input vector, $\boldsymbol{x}$. There is, therefore an increase in error for each input $\boldsymbol{x}$ for which the output of the $f_{opt}(\boldsymbol{x})$ differs from the overall optimum filter.

As f_{opt} and y_r depend only on the weight $|\boldsymbol{x}|$, they can be written as $f_{opt}(|\boldsymbol{x}|)$ and $y_r(|\boldsymbol{x}|)$. Since f_{opt} is optimal with respect to weight-based filters, its MAE cannot exceed the MAE of y_r, which means that rank-order filters are poorer than optimal weight filters. Indeed, $f_{opt} = y_r$ if and only if $P(Y = 1|\ w)$ ³$0.5$ for w ³ r and $P(Y = 1|\ w) < 0.5$ for $w < r$.

Now suppose the ideal and observed images possess the following weight-monotonic property:

$$P(y=1||\boldsymbol{x}|=i) \text{ ³ } P(y=1||\boldsymbol{x}|=j) \text{ for } i>j \quad (11)$$

Then $f_{opt} = y_{r_{opt}}$, where

r_{opt} = min value of r for which $P(y=1||\boldsymbol{x}|=r) >0.5$.

The weight-monotonic property states loosely that the more black pixels in the observation window, the more likely it is that the ideal pixel at the window centre is black. The model is not unreasonable for ideal images in which the micro-geometry is somewhat random and the noise is white and symmetric. Simulation show that these assumptions hold for restoration type problems where the noisy and ideal images have similar pixel values, but they do not hold for inverted or edge detected images. Rank order filters would not be applicable for the latter type of images.

If $r_{opt} = (|B|+1)/2$ then the optimum rank order filter is the median, otherwise it is some other rank.

The MAE of the optimum filter is then given as,

$$MAE < I_0, y_{opt}(I) >= \begin{bmatrix} \sum_{|\mathbf{x}|=0}^{|\mathbf{x}|=r_{opt}-1} P(|\mathbf{x}|)P(y=1 \mid |\mathbf{x}|) \\ + \sum_{|\mathbf{x}|=r_{opt}}^{|\mathbf{x}|=|B|} P(|\mathbf{x}|)P(y=0 \mid |\mathbf{x}|) \end{bmatrix} \quad (12)$$

The difficulty in this general approach to filter design is in obtaining a good estimate of the conditional and prior probabilities $P(y=1|\boldsymbol{x})$ and $P(\boldsymbol{x})$ respectively for each value of $\boldsymbol{x}$. In restricting the class of functions to that corresponding to rank order filters, a smaller set of conditional and prior probabilities $P(y=1||\boldsymbol{x}|)$ and $P(|\boldsymbol{x}|)$ must be estimated. This is carried out through the collection of observations of a representative training sequence.

OPTIMUM WEIGHTED MEDIAN FILTER DESIGN

The design of the optimum weighted median filter within a window B, reduces to the problem of determining the pixel weighting, W, for which the MAE is a minimum. This problem may be placed in the context of a difference filter, $D(I)$.

As the weighted median is a self dual operator, it treats foreground and background pixels equally, i.e. if a black pixel switches to white when a given number of its neighbours are white, then a white pixel with the same number of black neighbours must switch to black. Therefore, only two quantities with the filter window influence its output, $\boldsymbol{x}_c$, the value of the pixel at the centre of the window, and $|\boldsymbol{x}_{c'}|$ the sum of the neighbouring pixels which have the opposite value. The centre weighting can be directly related to the number of neighbouring pixels, of opposite value, $|\boldsymbol{x}_{c'}|$ required to cause the centre pixel $\boldsymbol{x}_c$ to switch its value from **0** to **1** or vice versa.

For simplicity let $d=D(I)$. Then $P(d=1||\boldsymbol{x}_{c'}|)$ is the probability that $\boldsymbol{x}_c$ will switch value when $|\boldsymbol{x}_{c'}|$ of its neighbours have the opposite value. Similarly $P(d=0||\boldsymbol{x}_{c'}|)$ is the probability that $\boldsymbol{x}_c$ will remain unchanged under the same conditions. The prior probability $|\boldsymbol{x}_{c'}|$ is given by $P(|\boldsymbol{x}_{c'}|)$.

By the same arguments as the rank order filter and as a result of the summation within the weighted median filter the difference filter $D(I)$ is increasing with $|\boldsymbol{x}_{c'}|$. Assuming that the weight -monotonic property holds, then the probability that a pixel will switch state increases monotonically with the number of neighbours it has with the opposite value.

The optimum differencing filter $D_{opt}(I)$ is determined by d'_{opt} the minimum value of $|\boldsymbol{x}_{c'}|$ for which $P(d=1||\boldsymbol{x}_{c'}|) > 0.5$.

Similarly the total MAE is given by

$$MAE < I_0, D_{opt}(I) >= \begin{bmatrix} \sum_{|\mathbf{x}_{c'}|=0}^{|\mathbf{x}_{c'}|=d'_{opt}-1} P(|\mathbf{x}_{c'}|)P(y=1 \mid |\mathbf{x}_{c'}|) \\ + \sum_{|\mathbf{x}_{c'}|=d'_{opt}}^{|\mathbf{x}_{c'}|=|B|} P(|\mathbf{x}_{c'}|)P(y=0 \mid |\mathbf{x}_{c'}|) \end{bmatrix} (13)$$

The conditional and prior probabilities may be estimated from observations of representative training images. Figure 2(a) shows an image containing very thin text. The probability estimates (b) show that a value of $d'_{opt} =7$ gives the optimum weighted median. This means that the pixel at the centre of the mask will switch state if 7 or 8 of its neighbours have the opposite value and this corresponds to a weight of $W=5$. In contrast applying the standard median results in the image shown in Figure 2(c) and it destroys most of the text. The result of applying the optimum weighted filter is shown in Figure 2(d) and it can be seen that most of the text is preserved. The filters with weights on either side of the optimum,i.e. $W=3$ *and* 7 give very poor results, which suggests that the selection of the optimum weight is critical.

Figure 2(e) shows an overview of the algorithm used to implement the optimum weighted median filter. It can be seen that the sorting stage has been replaced by a linear convolution with mask $B(0,0)=5$ and $B(k, l)=1$ for all k¹$0$ or l¹$0$. This is followed by threshold operation at $(W+9)/2= 7$.

ERROR ESTIMATES

Clearly in practical situations the ideal image is not available. Where the process is repeatable such as transmission error it is possible to transmit a number of test images to build the training set. In other cases it is necessary to model the noise in some way. A vitally important part of these methods lies in using the correct size of training set. Where a training set is too small an additional error known as the precision error is introduced. This quantity is a random function as it depends on the training set.

For a basic *3 x 3* filtering window, the *9* input variables means that there are $2^{2^9} \approx 10^{154}$ different logic functions which may be implemented. For larger windows the number of functions grows rapidly. The training set required to estimate the conditional probabilities in order to determine the optimum function out of all these combinations may be impossibly large. In this work the number of functions has been reduced to the set of functions which implement rank order, morphological and weighted median functions. The optimum parameters for these filters have been obtained from a training set of a few test images with negligible estimation error. A full discussion of the problems of estimation error in this type work is given in [7].

CONCLUSIONS

Many researchers and engineers reject morphological and other non linear image processing because of the need to introduce additional hardware or software functions to their system. This work has shown how a useful subset of non linear operations may be implemented using linear image processing tools to obtain median, rank, morphological and weighted median filters. Analysis and examples have been included to show the reader how to estimate the optimum filter parameters for these various types of filter. The work in this paper has been limited to binary image processing but it is also possible to use a similar framework for the implementation of greyscale functions. This will be the subject of a future paper.

REFERENCES

[1]Serra, J., Image Analysis and Mathematical Morphology, Academic Press, London, 1982.

[2] Bovik A. C., Huang, T. S. and Munson, D. C. Jr., "A generalization of median filtering using linear combinations of order statistics," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-31, pp. 1342-1350, Dec. 1983.

[3] Huber, P.J. Robust Statistics, Wiley, New York, 1981.

[4]Bednar J. B. and Watt T. L., "Alpha-trimmed means and their relationship to median filters," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-32, pp. 145-153, Feb. 1984.

[5] Yin, L., Yang, R, Gabbouj, M., Neuvo, Y., "Weighted Median filters: a tutorial", IEEE Trans on Circuits and Systems-II: Analog and Digital Signal Processing, vol 43, 157-192, March 1996.

[6] Marshall, S. 'Optimum non linear document restoration through linear greyscale operators', submitted to IEE Journal Vision Image and Signal Processing VISP, April 2001.

[7]Dougherty, E.R, , Loce R, 'Precision of morphological-representation estimators fror translation invariant binary filters: Increasing and non increasing", Signal Processing, 40, 129-145 (1994).

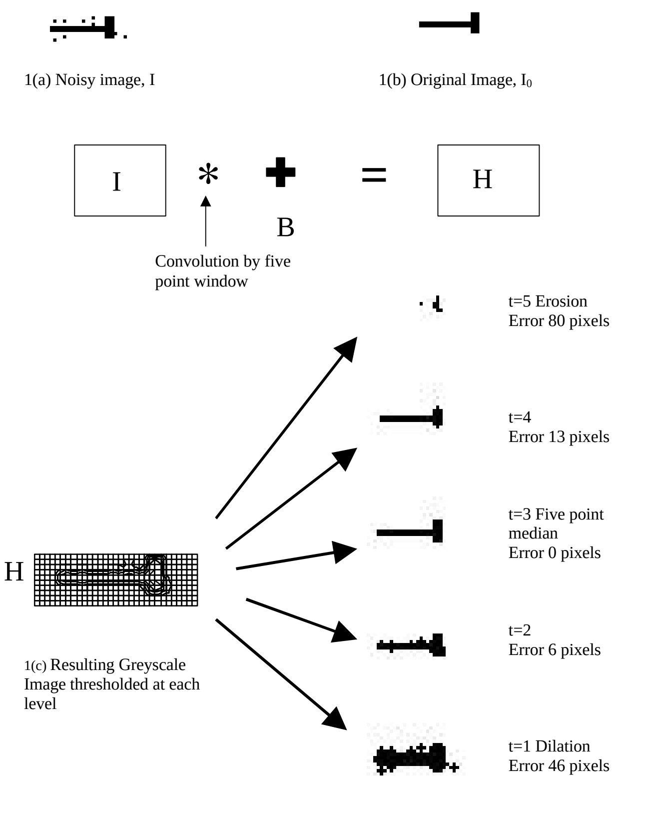
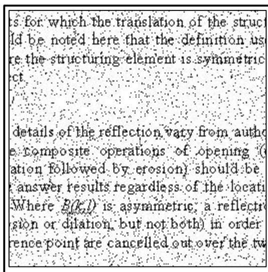


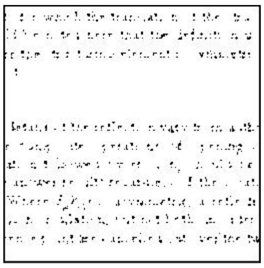
Figure 1
Convolution of noisy image with binary window then thresholded to give 5 output images including the erosion (t=5), dilation(t=1) and median (t=3).


s for which the translation of the struc
ld be noted here that the definition us
re the structuring element is symmetric
ct
details of the reflection vary from auth
e composite operations of opening (
ation followed by erosion) should be
answer results regardless of the locati
Where B(k,l) is asymmetric, a reflecti
sion or dilation, but not both) in order
rence point are cancelled out over the tw

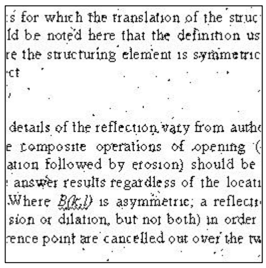
(a) Original noisy image

|xc'|	P(|xc'|)	P(d=0,|xc'|)	P(d=1| |xc'|)	D(x)
0	0.45	1.00	0.00	0
1	0.22	1.00	0.00	0
2	0.09	1.00	0.00	0
3	0.07	0.99	0.01	0
4	0.05	0.97	0.03	0
5	0.04	0.91	0.09	0
6	0.05	0.89	0.11	0
7	0.01	0.22	0.78	1
8	0.02	0.01	0.99	1

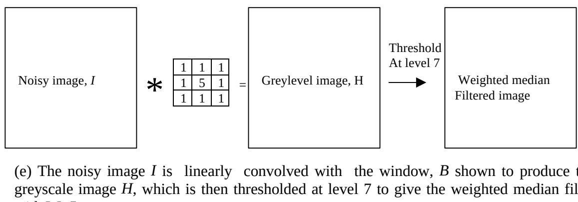
(b) Probability estimates



(c) Standard median filter


s for which the translation of the struc
ld be noted here that the definition us
re the structuring element is symmetric
ct
details of the reflection vary from auth
e composite operations of opening (
ation followed by erosion) should be
answer results regardless of the locati
Where B(k,l) is asymmetric, a reflecti
sion or dilation, but not both) in order
rence point are cancelled out over the tw

(d) Weighted median filter W=5.



(e) The noisy image *I* is linearly convolved with the window, *B* shown to produce the greyscale image *H*, which is then thresholded at level 7 to give the weighted median filter with W=5.

Figure 2
This image (a) containing very thin text is almost destroyed by the median filter(c), but the weighted median filter (d) with W=5 (equivalent to $|x_c|$=7) preserves much of the detail, (e) shows how the weighted median filter was implemented,