Reprinted from

Ninth International Symposium

Machine Processing of

Remotely Sensed Data

with special emphasis on

Natural Resources Evaluation

June 21-23, 1983

Proceedings

Purdue University
The Laboratory for Applications of Remote Sensing
West Lafayette, Indiana 47907 USA

LOCAL ADAPTIVE ENHANCEMENT: A GENERAL DISCUSSION AND FAST IMPLEMENTATIONS

E.C. DRISCOLL, JR., C. WALKER

International Imaging Systems
Milpitas, California

I. ABSTRACT

This paper will generally present a class of image enhancement operators that have been referred to as "Local Adaptive". It will contrast local and global operators and describe the cost tradeoffs associated with each. It will briefly summarize previous implementations of local adaptive enhancement. Finally, it will conclude by demonstrating two classes of local operators and their fast implementation on a general purpose image processor.

II. INTRODUCTION

When a user applies an enhancement operator, the intent may be to produce a visually pleasing result, or it may be to enhance the apparent information content of the image. These two intents may actually conflict, i.e. information extraction may produce a less visually pleasing image. From the point of view of information extraction, we can look upon image data as a matrix of small and large scale noise with bits of information unpredictably scattered throughout. Some users will want to filter out the extraneous noise and glean from the image those bits of useful information. In many cases, this means enhancing medium to small scale artifacts such as edges and textures while suppressing large scale shading and small scale random noise.

Global enhancements, such as a contrast stretch, change all the pixels in an image based only on global information. In other words, the same transformation is performed on each pixel in the image. In some cases, the large scale noise in the image, e.g., sensor shading or large changes in global image intensity, can interfere with the construction of an optimum global enhancement. Local adaptive operators enhance an image by transforming individual pixel values based on the characteristics of their surrounding neighborhoods. This can be expressed in the following form:

$$\mathbf{output_{x,y} = F\ [input_{x,y}\ ,\ G\ [H\ [input,x,y,w]\ ]\ ]} \qquad (1)$$

where:

output$_{x,y}$ and **input$_{x,y}$** are pixel values at location x,y, for the output and input images respectively.

F [ ] defines the local adaptive operator.

G [ ] produces a fixed set of summary values for sets of pixels.

H [ ] is the set of all pixels in the local window centered about x,y.

w is a constant that defines the local window size.

The global enhancement can now be seen as a simplification of equation (1) where w is set to a very large number (greater than the size of the image). In this case, the function H includes all pixels in the image regardless of x and y and therefore the function G will always return the same values and dictate the same transformation.

In the local adaptive case, each pixel can potentially undergo a different transformation, because each pixel has a unique local neighborhood. If an image is made up of both light and dark areas, i.e. its pixels represent more than one population, then global operators may reduce the information content of the image rather than increase it. This is because a given intensity level in one area may have a different meaning that the same intensity level in a different area. Global operators do not maintain this distinction; they treat all pixels with the same intensity level as the same, regardless of their surroundings. On the other hand, local operators try to adapt to the neighborhood around each pixel, enhancing locally meaningful relationships that a global operator might ignore. For example, in the dark areas of an image we might want to brighten the data to better distinguish features, while in the light areas of the same image we might want darken the data to again enhance local features. An operator which can do different things based on local conditions has been called "Local Adaptive".

III. PARAMETRIC VERSUS NON-PARAMETRIC LOCAL ADAPTIVE OPERATORS

Parametric operators generate enhancements by acting on locally derived summary statistics. In other words, they calculate the mean and standard deviation of a finite window around each pixel and then apply a transformation based on those summary statistics. Nonparametric operators actually calculate the histogram of each finite, local window, and then transform the center pixel based on that local histogram.

The form of function G in equation (1) can vary. Wallis[1] described a parametric operator based on a locally derived mean and standard deviation. In other words:

$$\mathbf{G}\ [\mathbf{W}_{x,y}] \Longrightarrow \mathbf{MEAN}_{x,y} \text{ and } \mathbf{STDEV}_{x,y} \tag{2}$$

where:

$W_{x,y}$ is the set of pixels in the window about x,y or simply H [input,x,y,w].

and

$$\mathbf{F}\ [\mathbf{input}_{x,y}, \mathbf{MEAN}_{x,y}, \mathbf{STDEV}_{x,y}] \Longrightarrow \tag{3}$$

$$\mathbf{output}_{x,y} = \mathbf{MEAN'} + (\mathbf{STDEV'} / \mathbf{STDEV}_{x,y}) * (\mathbf{input}_{x,y} - \mathbf{MEAN}_{x,y})$$

where MEAN' and STDEV' are the desired characteristics of the output.

In the described implementation, these local moments were not calculated at each location; rather, they were determined for non-overlapping areas and the actual transformation was generated for each discrete location by a bilinear interpolation scheme. It has since been recognized[7] that equation (3) can be implemented at high speed by the use of convolution operators. In this light, equation (3) can be approximated by:

$$\mathbf{output}_{x,y} = \mathbf{MEAN'} + \mathbf{SDK} * \frac{\mathbf{input}_{x,y} - \mathbf{LOW}\ [\mathbf{W}_{x,y}]}{\mathbf{LOW}\ [\ |\mathbf{input}_{x,y} - \mathbf{LOW}\ [\mathbf{W}_{x,y}]|\ ]} \tag{4}$$

where

LOW is a low pass convolution (that yields local mean)

MEAN' is a desired output mean

SDK is a constant that is related to the desired output standard deviation

A similar formulation has been proposed by Narendra and Fitch[2].

A non-parametric approach involves the calculation of local area histograms. In this paper, we describe a function G in equation (1) that returns a rank for the central pixel relative to its neighborhood. Specific types of rank filters have been described in Huang et al[3], Nakagawa and Rosenfeld[4], Mannos and Wolfe[8], Tyan[5] and Heygster[6]. They include MIN, MAX and MEDIAN filters. Our implementation uses the following equation:

$$\mathbf{G}\ [\mathbf{W}_{x,y}] \Longrightarrow \mathbf{RANK}\ [\mathbf{input}_{x,y}, \mathbf{W}_{x,y}] \tag{5}$$

and

$$\mathbf{F}\ [\mathbf{input}_{x,y}, \mathbf{RANK}_{x,y}] \Longrightarrow \tag{6}$$

$$\mathbf{output}_{x,y} = \mathbf{k} * \mathbf{RANK}\ [\mathbf{input}_{x,y}, \mathbf{W}_{x,y}]$$

where

RANK returns the CDF value for that location and surrounding window population

k is a radiometric control constant that maps the output values onto a desired range

This amounts to a local adaptive histogram equalization (LAHE). It can be adapted to perform any histogram based transformation (e.g. normalization, hyperbolization) by modifying the scaling transform to be appropriately non-linear.

IV. IMPLEMENTATIONS IN AN IMAGE PROCESSOR

One of the major factors that has limited the widespread application of local adaptive filtering has been the enormous computational expense involved. For example, to perform the above mentioned LAHE operator on a 512x512 image using a 63x63 window theoretically involves 262,144 separate histograms of populations of 3969 pixels. However, equations (4) and (6) can be economically implemented in a general purpose image processor. We have used a Model 75 Image Processor, manufactured by International Imaging Systems. Briefly, it consists of a pool of refresh memories whose data is directed through lookup tables, scroll, and zoom hardware into any of three high-speed processing pipelines. These pipelines allow full precision summing of any or all data streams from the refresh memories, followed by a lookup table transformation of the results. The resultant three data streams can be directed to digital-to-analog converters which drive red, green, and blue inputs of a color display monitor. Alternatively, the data can pass back through a feedback path to a higher precision processing unit and on to the refresh memories, again through a final scaling lookup table. The pipelines are frame-rate synchronous, producing a new resultant image every 30th of a second, resulting in a compute time of 127ns per pixel. The implementations in the image processor are therefore very fast. Execution times are dependent on the window size chosen and other optional parameters but they range from a few seconds to a few minutes for the entire 512x512 image.

The RANK function of equation (6) is implemented with the scroll, image summation, lookup table and feedback path elements of the image processor. The basic algorithm computes the sum of:

$$\mathbf{SIGN}\ [\mathbf{input}_{x,y} - \mathbf{input}_{i,j}]$$

for all elements $\text{input}_{i,j}$ in $W_{x,y}$.

where:

$$\mathbf{SIGN}\ [x] = \begin{cases} 1 & \text{if } x \text{ is positive} \\ 0 & \text{if } x \text{ is zero} \\ -1 & \text{if } x \text{ is negative.} \end{cases}$$

This sum after appropriate scaling, is equivalent to the average of two adjacent CDF values, producing a more balanced result than the raw CDF. The SIGN function can be computed and accumulated for the entire image in one frame time. Since it must be computed once for each element in the window, it follows that the algorithm will take one frame time per window element to complete (plus an additional frame time to scale the resulting sum to positive values).

In an attempt to achieve higher speed, we have explored techniques for subsampling the local window, thereby accelerating the collection of local summary statistics or histograms. This amounts to modifying the function H in equation (1). We have found that sampling only those local neighbors within w pixels of x,y that lie on radial "spokes" produces visually identical results in less time. This is particularly useful because the number of local neighbors to be examined (and therefore the execution time) rises linearly with window size in the radial spoke sampling, while it rises by the square of window size in an unsubsampled neighborhood. Also, the radial spoke sampling concentrates attention on nearby neighbors in preference to distant neighbors. Therefore, although it can be found to introduce artifacts, they are only associated with large pathological situations; isolated extreme values are more properly handled.

The LAHE operator described in equation (6), like global histogram equalizations, can prove to be too harsh. This is particularly the case where, for example, 7 bits of signal and 1 bit of noise are stored in an 8 bit image. The ranking algorithm will separate pixels by their noise component in areas where the signal component is flat (uniform areas). Our image processor implementation allows the user to control the enhancement of pure noise by effectively adding a "fuzz" factor to the ranking algorithm. In other words, the user can define a threshold such that neighboring pixels that differ by less than that threshold are considered "equal" and ranked equally. The image processor implementation allows the introduction of this threshold without impacting execution time.

V. EXAMPLES

Figures one thru nine show the effect of both the non-parametric and parametric local adaptive operators, on a medical x-ray, at various window sizes. Figures 10 thru 13 show the operators on Landsat MSS data. Of particular interest is the effect of the difference threshold, as indicated in figures 11 and 12.

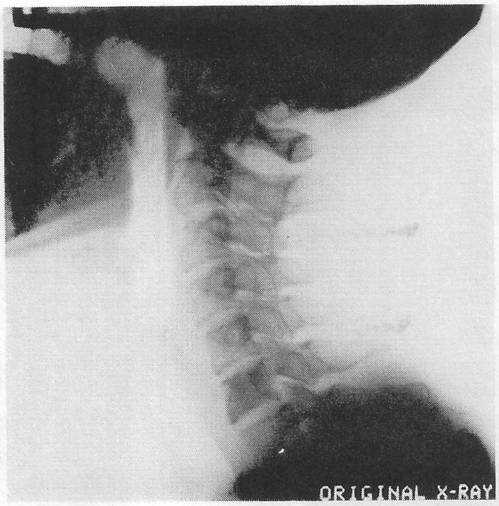

Figure 1. Original X-ray

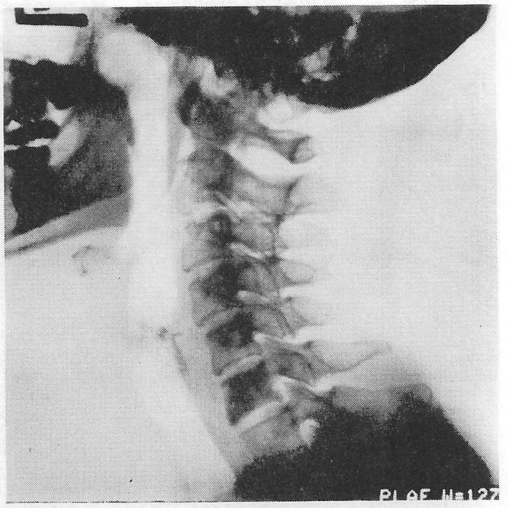

Figure 2. Parametric, Window=127

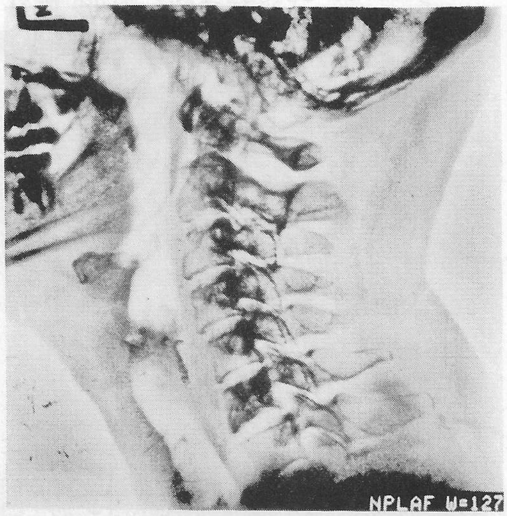

Figure 3. Non-parametric, Window=127

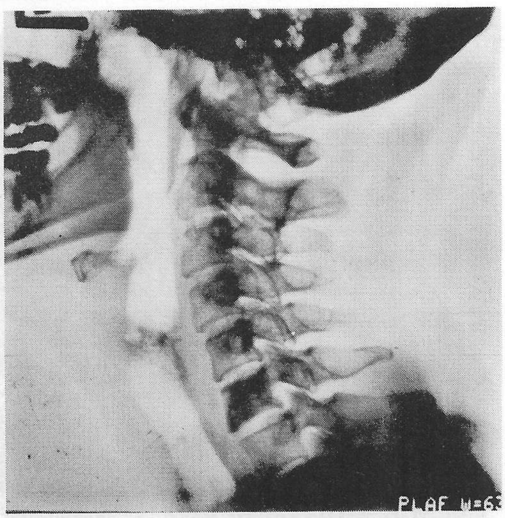

Figure 4. Parametric, Window=63

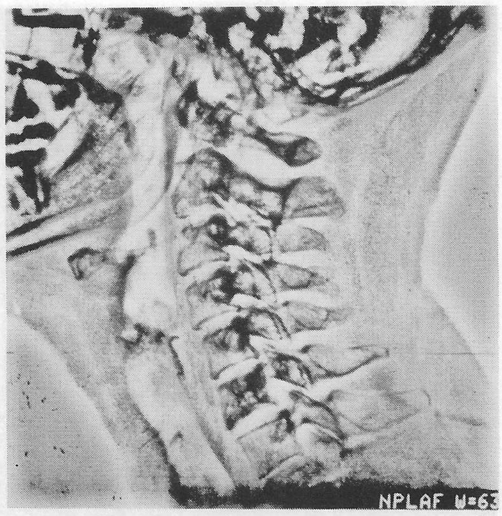

Figure 5. Non-parametric, Window=63

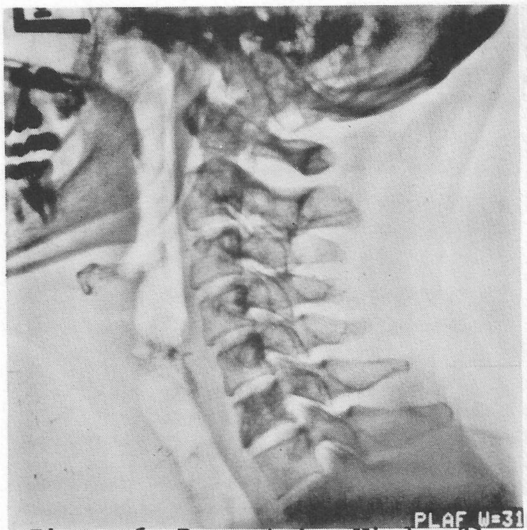

Figure 6. Parametric, Window=31

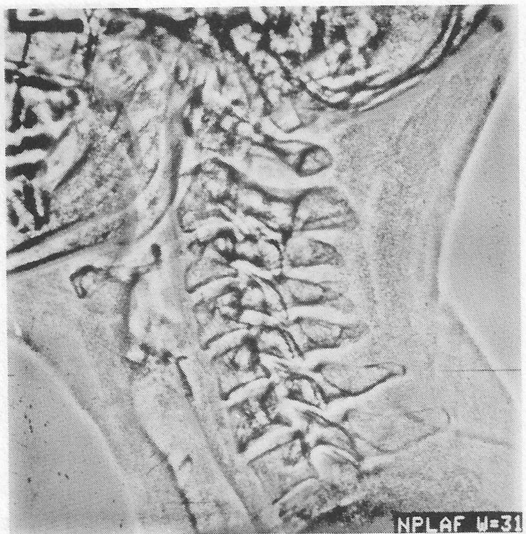

Figure 7. Non-parametric, Window=31

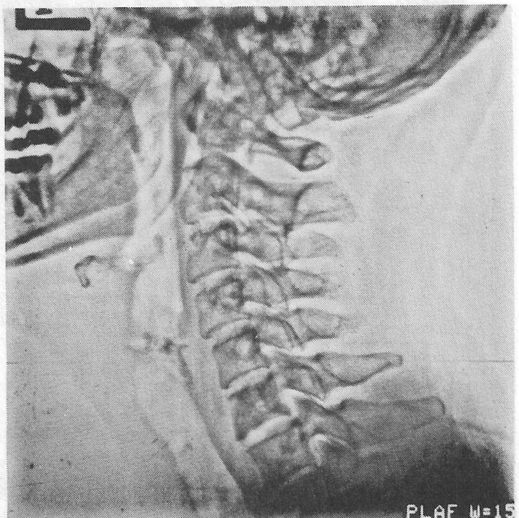

Figure 8. Parametric, Window=15

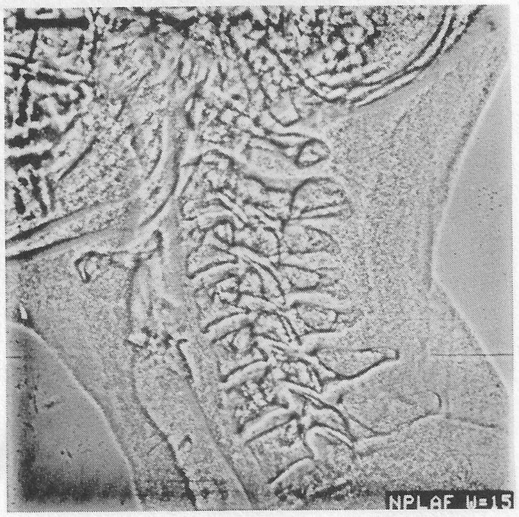

Figure 9. Non-parametric, Window=15

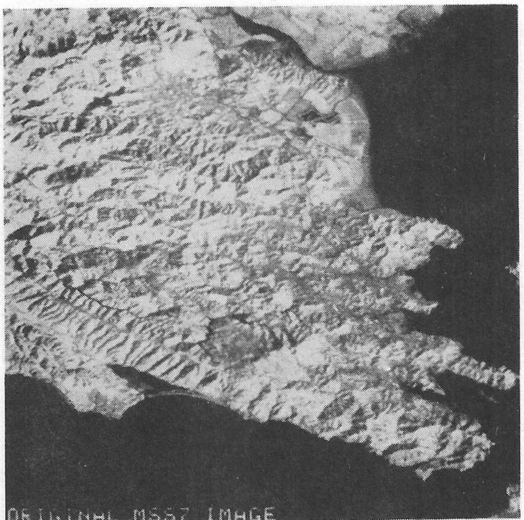

Figure 10. Original Landsat Band 7

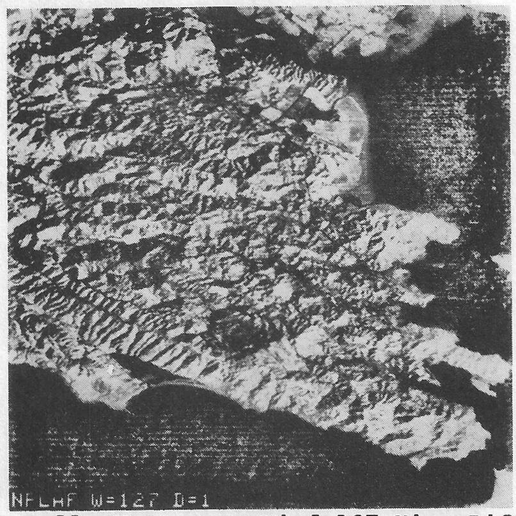

Figure 11. Non-para. Wind=127 Min. Diff=1

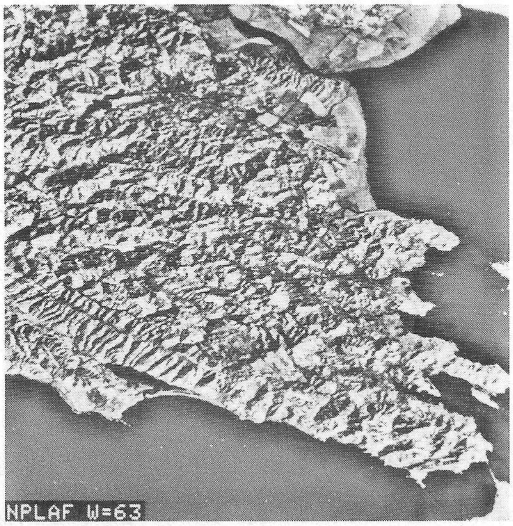

Figure 12. Non-para. Wind=63 Min. Diff=4

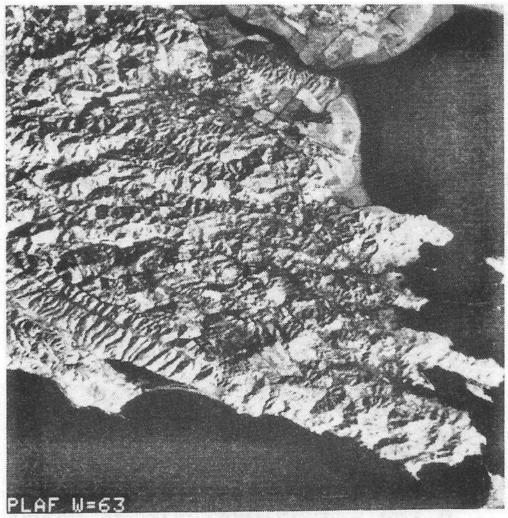

Figure 13. Parametric Window=63

Bibliography

[1] Wallis, R., "An approach to space variant restoration and enhancement of images" in Proc. of Symp. on Current Mathematical Problems in Image Science, Naval Postgraduate School, Monterey, CA, Nov. 1976

[2] Narendra, P.M. and R.C. Fitch, "Real-time adaptive contrast enhancement", in IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI-3, No. 6, Nov. 1981

[3] Huang, T.S., G.J. Yang and Y.G. Tang, "A fast two-dimensional median filter implementation", Proc. 1978 Conf. on Pattern Recognition and Image Processing, Miami Beach, 1980

[4] Nakagawa, Y. and A. Rosenfeld, "A note on the use of local Min and Max operations in digital picture processing", IEEE Trans. of Systems, Man and Cybernetics, Vol. SMC-8, 1978

[5] Tyan, S.-G., "Median filtering: Deterministic properties" in Two-dimensional Digital Signal Processing II (T.S. Huang ed.) Top. Appl. Physics, Vol. 43, Springer-Verlag, Berlin, 1981

[6] Heygster, G., "Rank filters in digital image processing", Computer Graphics and Image Processing 19, pp 148-164, Academic Press, 1982

[7] Driscoll, T. and C. Walker, "Evolution of Image processing algorithms from software to hardware", Proceedings of SPIE, Vol. 271

[8] Mannos, J. and G. Wolfe, "Fast median filter implementation", Proceedings of SPIE, Vol. 207

Authors

Edward C. Driscoll. Edward Driscoll is Applications manager at International Imaging Systems in Milpitas, California, where he is responisble for the development of new applications software in a variety of areas including regional planning and environmental analysis. He received his B.A. (1974) from the University of Pennsylvania and Masters degrees in architecture and regional planning/landscape architecture from the Harvard Graduate School of Design. He is presently completing work toward a PhD. in earth sciences at Stanford University.

Christopher C. Walker. Chris Walker is manager of Systems software at International Imaging Systems, where he is resposible for design and implementation of software for turnkey image processing systems. Prior to this position, he was responsible for implementation of various image processing algorithms. He has completed three years of study towards an A.B. in computer science at the University of California, Irvine and Berkeley.