Adaptive Box Filters for Removal of Random Noise from Digital Images

Eric M. Eliason and *Alfred S. McEwen*
U. S. Geological Survey, 2255 N. Gemini Drive, Flagstaff, AZ 86001

ABSTRACT: We have developed adaptive box-filtering algorithms to (1) remove random bit errors (pixel values with no relation to the image scene) and (2) smooth noisy data (pixels related to the image scene but with an additive or multiplicative component of noise). For both procedures, we use the standard deviation (σ) of those pixels within a local box surrounding each pixel, hence they are adaptive filters. A data point is considered to be a bit error if it deviates from the box mean by more than 1.0 to 2.0 σ, and it is replaced by the box mean. Even very minor bit errors are removed from low-variance areas, but valid data along sharp edges and corners are not replaced (i.e., in high-variance areas). A series of two or three filters with decreasing box sizes can be run to clean up extremely noisy images and to remove bit errors near sharp edges. Our second filter, for noise smoothing, is identical to the "sigma filter" of Lee (1983a) except that we use the local (adaptive) σ rather than a fixed σ. The filter averages only those pixels within the box that have intensities within 1.0 to 2.0 σ of the central pixel. This technique effectively reduces speckle in radar images without eliminating fine details.

INTRODUCTION

In this paper we describe adaptive box-filtering algorithms for the removal of random noise from digital images. Our techniques are simple and rapid, yet they have advantages over previously described techniques, especially in terms of preserving the full resolution and detail in the data. The highest possible image resolution has proven essential for many scientific interpretations, and our goal is to preserve the detail of the entire scene while removing or reducing the noise.

We present algorithms for two very different types of random noise: (1) single-pixel, random bit errors or "salt-and-pepper" noise, for which the pixel values have no relation to the image scene; and (2) noisy data, or pixels whose intensity is related to the image scene but which have an additive or multiplicative component of noise. We use "noisy pixels" or "noisy images" with reference to both types of noise, but we use "noisy data" only for the second type. In the first case our goal is to eliminate the bit errors by replacing the pixel value with the mean value of surrounding valid pixels, and in the second case we wish to smooth the noisy data. In this paper we concentrate on the first problem, but we also present a technique for the second problem that minimizes the blurring of scene detail.

The occurrence of bit errors is especially severe for images acquired by some planetary spacecraft missions such as Mariner and Viking, because the telemetry occurred over very large distances. The bit errors increase with telemetry data rate; a higher bit-error rate must be accepted when a high data rate is needed, for example, when acquiring as much high-resolution imaging as possible during the closest approach to a planet (*cf.* Danielson *et al.*, 1975). About 80 percent of the Mariner 10 images of Mercury were acquired at a data rate that was high for that mission (117.6 kbit/s), and more than 10 percent of the values in these images consist of bit errors. In addition, data are often "dropped" or lost in various ways. For example, Viking Orbiter images were initially recorded on 7-track magnetic tape recorders on the spacecraft, then played back to Earth one track at a time. Loss of a track during telemetry results in missing bits at 7-pixel intervals across image scan lines. Dropped data are usually replaced by the average of surrounding pixels, but, if the surrounding pixels contain many bit errors, this procedure is not satisfactory. Some of the highest resolution sequences acquired over portions of Mars by the Viking Orbiters consist of as much as 50 percent bit errors or dropped data, and the "raw" image appears unusable (see Figure 2). However, with the techniques described here, we can produce images that appear blemish free with no apparent loss of resolution.

Noise is inherent in electrical circuits and is introduced from external sources. Noisy data are a combination of signal and noise, as distinguished from random bit errors that are entirely noise. We are particularly interested in reducing speckle noise in synthetic aperture radar (SAR) images because the spacecraft Magellan will map Venus with radar and will return more digital data than from all previous spacecraft missions to the planets (except Earth) combined. We have tested the noise smoothing algorithm of Lee (1983a, 1983b) on SAR images, and we have found that an adaptive modification to his technique (using the local σ) produces promising results.

The filter programs described here are part of the PICS (Planetary Image Cartography System) software package developed at the U. S. Geological Survey in Flagstaff, Arizona. PICS is available to the NASA-funded planetary research community. The software runs on a VAX or MicroVAX with the VMS operating system. PICS emphasizes cartographic aspects of image processing (Batson, 1987; Edwards, 1987), but it includes a wide range of image-processing algorithms.

"Box" filters consider the values within a rectangular box centered on each pixel, and have been in use since the early 1970s at the Jet Propulsion Laboratory, Pasadena, and at the U. S. Geological Survey, Flagstaff (Soha *et al.*, 1975; Eliason and Soderblom, 1977). The box-filtering algorithm has been described in detail by Seidman (1972), McDonnell (1981), and Schowengerdt (1983). The main advantage of box-filtering is its speed. It requires only four add/subtracts to compute the sums for each box, and the speed is nearly independent of box size. The box-filtering algorithm is ~10 times faster than "direct filtering" (summing each value in each box) with a 5 by 5 box, and ~4000 times faster than direct filtering with a 101 by 101 box. Furthermore, box-filtering is ~15 times faster than the fast Fourier transform (FFT) for a 1024 by 1024 image, and it does not have the disk and main memory storage requirements of the FFT.

REMOVAL OF BIT ERRORS

Bit-error noise consists of discrete isolated pixel variations or "spikes" and gives an image a "salt-and-pepper" appearance. A bit-error pixel often has an intensity markedly different from its neighbors, and a simple "out-of-range" noise-removal method

PHOTOGRAMMETRIC ENGINEERING AND REMOTE SENSING,
Vol. 56, No. 4, April 1990, pp. 453–458.

0099-1112/90/5604–453$02.25/0

is most commonly used (e.g., Soha *et al.*, 1975; Chavez, 1980; Schowengerdt, 1983). In this technique, each pixel is compared with the average of the surrounding box, and, if the difference exceeds a specific threshold value, the pixel is replaced by the average of neighboring pixels. This technique is very effective for widely-scattered bit-error pixels in a relatively low contrast scene. However, if the bit errors are very dense, the box average is likely to be influenced by the bad values, which has two effects: (1) the bit errors are less likely to be identified as such and (2) the average may not be an appropriate value for replacing a bit error. Another problem with this technique is that valid pixels along high-contrast boundaries may be erroneously classified as bit errors and replaced by the average, resulting in degradation of the edges and elimination of thin lines and sharp corners. These problems may be eliminated by increasing the threshold value, but this leaves many bit errors uncorrected. Uncorrected bit errors may be especially annoying during later stages of image processing because they are made more prominent by edge-enhancement filters, color ratios, and other processing.

We have solved the problems associated with the standard noise-removal algorithm by using an "adaptive" filter that uses the local statistics (mean and standard deviation, σ) within each box to determine whether a pixel is classified as valid or invalid data. Adaptive filters have been previously described for smoothing noisy data (e.g., Frost *et al.*, 1981, 1983; Lee, 1980, 1981a, 1981b; Tom, 1985), but we wish to completely remove bit errors rather than to smooth the noise. An adaptive filter has also been described for edge enhancements (high-pass filters) that avoids the "ringing" artifact (Torres *et al.*, 1988).

Our bit-error removal technique is similar to the threshold method described above, except that the threshold value is defined as some multiple, C, (usually from 1 to 2) of the σ of each box. In low-variance regions the threshold value is small and even very low level bit errors are recognized, whereas in high-variance regions the threshold value is large so that valid data are not replaced. Only the most extreme bit errors are replaced in high-contrast areas. However, we can replace bit errors near sharp edges by running a series of filters with decreasing box sizes. The box-filtering algorithm allows several filters to be run in a reasonable period of time (about 43 seconds for each filter of a 512 by 512 image on a MicroVAX-II).

A different bit error removal technique was described by Rosenfeld and Kak (1982, p. 252). In their method, a pixel is defined as a bit error if it differs by more than a specific threshold value from a certain number ("most of") its neighbors. This method reduces the likelihood that valid points on edges or lines will be confused with bit errors. However, Lee (1983a, p. 265) presented a method that is identical with that of Rosenfeld and Kak (1982, p. 252) and he demonstrated that on very noisy images not all of the bit errors can be removed without destroying thin lines and blurring edges.

Another method commonly used to remove noise spikes without blurring edges is the median filter, in which each pixel is replaced by the box median value (e.g., Rosenfeld and Kak, 1982, p. 261). Although the median filter works well on a one-dimensional string of data, the two-dimensional median filter destroys thin lines and "clips" corners.

Our method avoids the problem of destroying thin lines and sharp corners when the threshold value is at least 1.5 σ. Consider a line or corner with a uniform gray level. In a normal distribution, 32 percent of the data differs from the mean by more than 1.0 σ, so no more than two pixels in a 3 by 3 box can be considered noise. A thin line or corner that passes through the central pixel of a 3 by 3 box must have at least three members in the box, so the line or corner will not be replaced unless C is less than 1.0. Likewise, a thin line or corner in a 5 by 5 box must occupy at least five pixels (1.3 σ), or seven pixels (1.5 σ) in a 7 by 7 box. We have encountered no problems with replacing valid data when using C of 1.5 or larger with box sizes up to 7 by 7, even with very high contrast scenes with sharp edges and corners (see Figure 1). A threshold of less than these limiting values ($C < 1.5$) may be used without replacing valid data if each box contains at least one bit error (which increases the computed σ compared with σ of the valid data) or if very sharp corners or lines are not present (see Figures 2 and 3).

All of the bit-error removal methods described above rely upon an assumption of spatial coherence in the image. It is assumed that the intensities of neighboring pixels are not completely independent and that a pixel is likely to have at least two neighbors with similar values. If the actual scene has a salt-and-pepper appearance with bright or dark features of only one to two pixels in size, these methods may eliminate these valid data points.

ADAPTIVE ALGORITHM FOR REMOVAL OF BIT ERRORS

Let $P(i,j)$ represent an integer sample of an image array P. Centered on $P(i,j)$ is a rectangular box of dimension $(2K+1)$ by $(2L+1)$, where K and L are integers. Further, let the valid data in the array be in the range MIN to MAX. The MIN and MAX parameters allow a delta function, D, to be defined:

$$D(i,j) = 0 \text{ if } P(i,j) < \text{MIN, or } P(i,j) > \text{MAX} \qquad (1)$$
$$D(i,j) = 1 \text{ if MIN} \leq P(i,j) \leq \text{MAX}$$

The delta function is used in the equations shown below to discriminate between valid and invalid pixel values. Typically for 8-bit integer images MIN=1 and MAX=255; zero indicates an invalid or empty pixel in the array. Invalid pixels can result from missing image data due to data dropouts or points that have been identified as bit errors and set to zero on output by the bit-error removal algorithm. In addition, MIN greater than 1 or MAX less than 255 may be used to exclude the most extreme noise from the box statistics, provided that all valid data values fall within the range defined by MIN-MAX.

Three sums must be computed for each box. The sum of valid points, $S(i,j)$, the number of valid points, $N(i,j)$, and the sum of the square of the valid points, $SS(i,j)$ are given by Equations 2, 3, and 4:

$$S(i,j) = \sum_{m=i-K}^{i+K} \sum_{n=j-L}^{j+L} D(m,n)P(m,n) \qquad (2)$$

$$N(i,j) = \sum_{m=i-K}^{i+K} \sum_{n=j-L}^{j+L} D(m,n) \qquad (3)$$

$$SS(i,j) = \sum_{m=i-K}^{i+K} \sum_{n=j-L}^{j+L} [D(m,n)P(m,n)]^2 \qquad (4)$$

The low-pass filter, LPF(i,j), is defined as the mean value of the box excluding invalid values. The replacement value, $R(i,j)$, is defined as the mean value of the box excluding invalid values and the central pixel:

$$\text{LPF}(i,j) = S(i,j)/N(i,j) \qquad (5)$$

$$R(i,j) = [S(i,j) - D(i,j)P(i,j)]/[N(i,j) - D(i,j)] \qquad (6)$$

When $P(i,j)$ is classified as a bit error, we use $R(i,j)$ rather than LPF(i,j) as the replacement value because the average does not contain the erroneous value of $P(i,j)$.

The variance of the box, $V(i,j)$, is given by Equation 7:

$$V(i,j) = [SS(i,j)/N(i,j)] - [S(i,j)^2/N(i,j)^2] \qquad (7)$$

and the standard deviation, $\sigma(i,j)$ is the square root of $V(i,j)$.

The bit error removal algorithm determines whether $P(i,j)$ is a bit error by comparing it with the statistics of the box. If $P(i,j)$ is considered a valid point, it will remain unchanged. If $P(i,j)$ is marked as a bit error, the algorithm will either zero the pixel

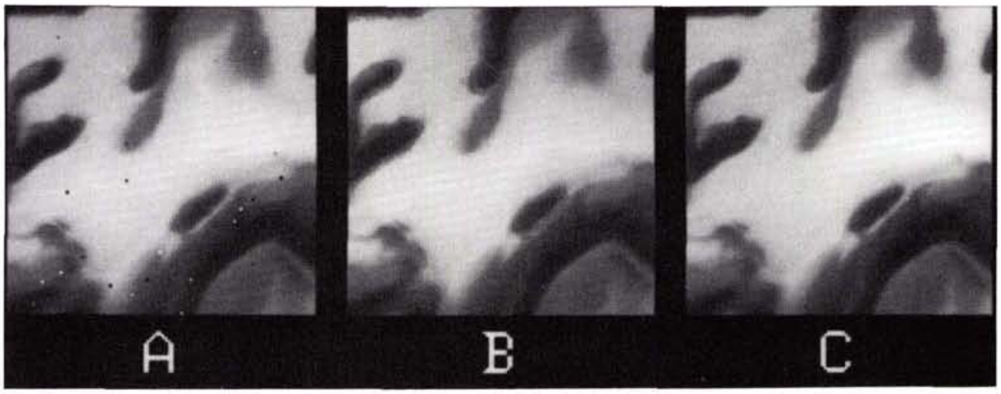

FIG. 1. Portion of Viking Orbiter image 765A25 of Mars' north polar ice cap. Image area with 100 lines and 100 samples. (A) Raw data. (B) Result of a 5 by 5 non-adaptive noise filter with a threshold value of 25. (C) Result of a 3 by 3 adaptive noise filter with a threshold value of 1.5 σ; note that two apparently noisy pixels that lie directly on an edge were not replaced.

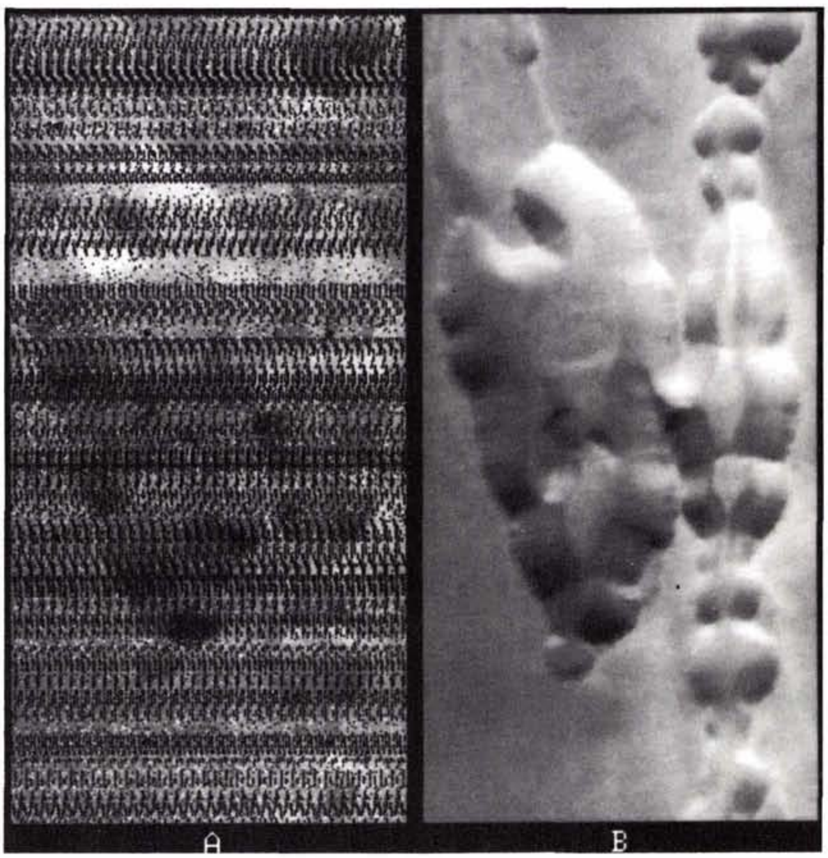

FIG. 2. Portion of Viking Orbiter image 927A04 of a chain of collapse pits called Tithonia Catena at latitude $-5.5°$, longitude 86°. Image area has 512 lines and 250 samples. Illumination is from the bottom left. (A) Raw data. (B) Result of three adaptive noise filters. The reseau marks, circular dark spots spaced about 120 pixels apart in (A) have been replaced by interpolation in (B).

or replace the pixel value with $R(i,j)$. The option to zero the data point is called the STDZ filter, and the option to replace the data point with the box-window average is called the STD filter. The STDZ option, described below, allows multiple applications of the bit-error removal algorithm for extremely noisy images.

$P(i,j)$ is classified as a bit error if the conditions in Equation 8 are met:

$$|P(i,j) - \mathrm{LPF}(i,j)| > C\sigma(i,j) \text{ and } |P(i,j) - \mathrm{LPF}(i,j)| > \mathrm{TOL} \quad (8)$$

where C is a constant (usually from 1 to 2, as described above) and the constant TOL is a minimum threshold value that must be exceeded. Both C and TOL are selected as inputs to the program. The parameter C represents the multiple of σ that $|P(i,j) - \mathrm{LPF}(i,j)|$ must exceed before $P(i,j)$ can be considered a bit-error

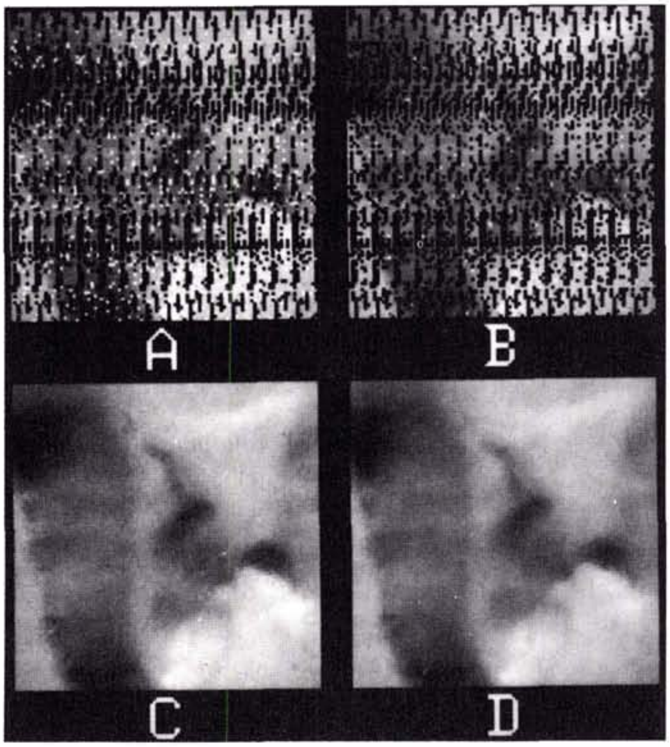

FIG. 3. Enlargement of 100 by 100 pixel area from central portion of image shown in Figure 2. (A) Raw data. (B) Result of 7 by 7 STDZ filter (see text). (C) Result of additional 5 by 5 STD filter to image in (B). (D) Result of additional 3 by 3 STD filter to image in (C).

candidate. (In practice we square both sides of the equations in Equation 8, which is more efficient for the computer than taking the square root.) The TOL parameter, typically in the range of 2 to 10 for 8-bit data, is used to avoid changing potentially valid data points in areas of very low variance or in areas where a pixel-scale roughness with brightness variations less than TOL might be considered valid data. In many cases TOL may be set to zero.

By using $\sigma(i,j)$, the filter becomes adaptive to the local scene contrast. Low-contrast areas of a scene will have a smaller standard deviation than high-contrast areas. Thus, according to Equation 8, a pixel value may be classified as a bit error with a relatively small value of $|P(i,j) - \mathrm{LPF}(i,j)|$ in a low-contrast area. As the local contrast increases, the discrimination of bit errors from real data becomes more difficult both visually and numerically. In an area of low contrast such as an area with little or no topographic or albedo variations, a noise spike is readily seen (see Figure 1). However, in areas of high contrast such as along topographic and albedo boundaries, it becomes more difficult to distinguish a noise spike from the high-contrast boundaries. When the box window is centered over one of these high-contrast boundaries, $\sigma(i,j)$ increases so that a larger value of $|P(i,j) - \mathrm{LPF}(i,j)|$ is required before $P(i,j)$ will be classified as a bit error. Thus, pixels that make up the high contrast boundary will not be classified as invalid data.

The STDZ option (to zero bit errors rather than to replace the pixel with the box mean) was created for two reasons. First, we do not wish to include bit errors (within a box but not the central pixel) in the mean used to replace invalid pixels. Second, larger boxes result in more successful classification of bit errors when other bit errors are present in the box, because of the improved statistics. However, the smaller boxes are best for replacing the invalid data with $R(i,j)$. Therefore, the solution to these problems is first to run the STDZ option with a relatively large box size, followed by the STD option with a smaller box. The standard sequence used in PICS for Viking Orbiter images is a 7 by 7 box STDZ filter followed by a 5 by 5 STD filter.

APPLICATION TO VIKING ORBITER IMAGES OF MARS

Our techniques are illustrated with two examples, both Viking Orbiter images of Mars. The first image is a part of the north polar residual ice cap (Figure 1). The contrast is very high between the bright white ice and the dark, ice-free lanes. The standard (nonadaptive) threshold bit-error filter replaces pixels along the sharp edges and degrades the image resolution (Figure 1b). If the threshold value is increased, most of the noise is not removed. In contrast, the adaptive filter (3 by 3 box, $C = 1.5$) replaces nearly all of the bit errors without degrading the valid data (Figure 1c).

The second example is an image that is very noisy and contains a great deal of dropped (zero) data (Figures 2 and 3). Only about 50 percent of the pixels contain valid data. A series of three adaptive filters is needed to clean up this image. We cannot begin with an interpolation filter to replace the missing data, because the box averages are strongly affected by the bit errors. Not only would the interpolated data appear unsatisfactory, but the nearby noisy pixels would be more difficult to recognize and remove by a subsequent bit-error filter because they would be correlated with the interpolated data. We also do not wish to replace the noisy pixels with the box mean during the first pass, for the same reason. We therefore used three filters: the first filter was a 7 by 7 STDZ to zero the noisiest pixels (Figure 3b), followed by a 5 by 5 STD to replace second-order noise and to interpolate the zeroed pixels (Figure 3c), and finally a 3 by 3 STD to remove third-order noise and noisy, pixels near edges (Figure 3d). A threshold value of 1.0 σ was used for all three filters.

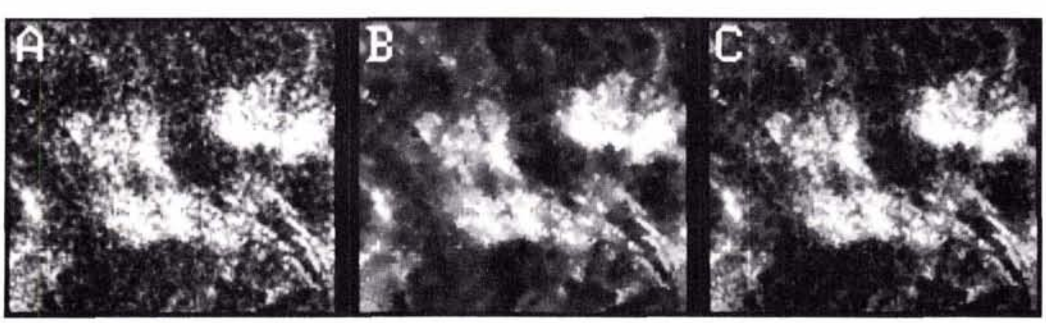

FIG. 4. Portion of Arecibo radar image of Venus. Image area with 114 lines by 128 samples. (A) Original image. (B) Result of 5 by 5 sigma filter with 1 σ threshold. (C) Result of 5 by 5 adaptive sigma filter with 1 σ threshold.

SMOOTHING NOISY DATA

Many techniques have been described for smoothing noisy data (see reviews by Pratt (1978) and Rosenfeld and Kak (1983)). The obvious disadvantage of a simple low-pass filter is that image sharpness and detail are reduced along with the noise. The use of a bit-error removal algorithm on noisy data is inappropriate because valid data would be eliminated. For example, speckle noise in SAR images is multiplicative (i.e., its σ increases with signal intensity; see Lee (1981b)), so bright areas are especially noisy, and use of a bit-error removal algorithm on SAR images tends to break up the continuity of bright thin lines and to corrode bright edges. Instead, we wish to smooth the noisy data to improve interpretability without degrading or eliminating valid data.

Elaborate algorithms have been developed to attempt simultaneous suppression of noise and preservation of fine image detail. However, Lee (1983a) concluded that a very simple method that he called the "sigma filter" is at least as effective and much more rapid than the more elaborate methods. The basic idea of the sigma filter is to replace each pixel value by the average of only those neighboring pixels that have an intensity within a fixed σ range of the center pixel. As a result, only those values that can be thought of as belonging to the same population are averaged. The sigma filter is identical to a low-pass filter with a limited valid intensity range (set by MIN and MAX), except that the valid intensity range varies with the intensity of the center pixel. Sigma is computed for the entire image, and the valid intensity range is defined by the value of the central pixel plus or minus a multiple (usually 1 to 2) of σ. Although this is a very simple modification of a commonly used filter, the improvement in the results is surprising. Lee (1983a) noted its advantages over a low-pass filter with a fixed MIN-MAX range:

> "(1) noise near edge areas will be smoothed without blurring the edge because only pixels on one side of the edge are included in the average; (2) subtle details of several pixel clusters and linear features of one to three pixels in width will be preserved since only those pixels and not the background are included in the average: (3) it will not create artifacts and will retain shapes, because no directional masks are used, unlike the algorithms of Nagao and Matsuyama (1979) and Lee (1981a); (4) it is computationally efficient, since only simple compare and fixed point add instructions are involved."

Lee (1983a) compared the computational efficiency of the sigma filter to several other algorithms. Compared with 1 unit of computation time for a sigma filter, the median filter requires 1.5 units of time, the gradient inverse filter (Wang *et al.*, 1981) requires 4 units of time, and the filter of Nagao and Matsuyama (1979) requires 11 units of time. However, the sigma filter is much slower than filters that can use the rapid box-filtering algorithm (Seidman, 1972; McDonnell, 1981). Because of the need to compare the value of the central pixel with the values of all other pixels in the box, it is not possible to use just 4 add/subtracts per summation. The computation time comparisons given above were for a 7 by 7 sigma filter, which was apparently done by "direct filtering," and a 7 by 7 low-pass filter using the rapid box-filtering technique would be about 20 times faster (McDonnell, 1981).

Lee (1983a, 1983b) discussed the application of the sigma filter to signal-dependent noise of speckles that occur in synthetic aperture radar (SAR) images (Goodman, 1976). He concluded that in many cases the sigma filter performs better and requires much less computational time than an earlier adaptive method (Lee, 1981b). We tested the sigma filter on a portion of an Arecibo image of Venus (Campbell *et al.*, 1976; see Figure 4). The speckle is greatly reduced and prominent edges and linear features have not been degraded. However, some of the low contrast edges and linear features have been degraded. The reason for this is easy to understand: all of the intensities in a relatively low contrast area are within 1.0 of the value of the central pixel, and these areas are given a simple low-pass filter.

In order to solve the problem of degrading fine detail in relatively low-contrast areas, we have modified Lee's sigma filter into an adaptive filter. We simply use $\sigma(i,j)$, computed for each box, rather than σ for the entire image. Previous adaptive filters for smoothing noisy data (Frost *et al.*, 1981, 1983; Lee, 1980, 1981a, 1981b; Tom, 1985) are adaptive in the sense that they apply greater smoothing to low-variance areas. These filtering methods are different from our method, and in fact they worsen the problem of degrading fine detail in low-contrast areas of the image. Our adaptive sigma filter (Figure 4c) preserves fine detail even in low-contrast areas, but it nevertheless reduces the speckle noise significantly. For scientific studies in which the fine detail may be important, we consider this adaptive sigma filter to be preferable to the standard sigma filter. However, it requires about 1.7 times more computation time.

CONCLUSIONS

We have presented two conceptually simple and relatively rapid adaptive filters for removing random noise from digital images. The first filter is designed to replace random bit errors, and the second is designed to smooth noisy data. The two filters can be combined into a single program for processing images with both random bit errors and noisy data. These filters are more successful than previous efforts for removing noise without degrading the fine image detail.

ACKNOWLEDGMENTS

This research was supported by NASA grant W-15,814 to the U. S. Geological Survey.

REFERENCES

Batson, R. M., 1987. Digital cartography of the planets: New methods, its status, and its future. *Photogrammetric Engineering and Remote Sensing*, Vol. 53, No. 9, pp. 1211–1218.

Campbell, D. B., R. B. Dyce, and G. H. Pettengill, 1976. New radar image of Venus. *Science*, Vol. 193, pp. 1123–1124.

Chavez, P. S., Jr., 1980. Automatic shading correction and speckle noise mapping/removal techniques for radar image data. *Radar Geology: An Assessment*. Jet Propulsion Laboratory, Pasadena, California, JPL Publication no. 80-61, pp. 251–262.

Danielson, G. E., K. P. Klaasen, and J. L. Anderson, 1975. Acquisition and description of Mariner 10 television science data at Mercury. *Journal of Geophysical Research*, Vol. 80, No. 17, pp. 2357–2393.

Edwards, K. E., 1987. Geometric processing of digital images of the planets. *Photogrammetric Engineering and Remote Sensing*, Vol. 53, No. 9, pp. 1219–1222.

Eliason, E. M., and L. A. Soderblom, 1977. An array processing system for lunar geochemical and geophysical data. *Proceedings Lunar and Planetary Science Conference*, 8th, pp. 1163–1170.

Frost, V. S., J. A. Stiles, K. S. Shanmugan, J. C. Holtzman, and S. A. Smith, 1981. An adaptive filter for smoothing noisy radar images. *Proceedings of the IEEE*, Vol. 69, No. 1, pp. 133–135.

Frost, V. S., M. S. Perry, L. F. Dellwig, and J. C. Holtzman, 1983. Digital enhancement of SAR imagery as an aid in geologic data extraction. *Photogrammetric Engineering and Remote Sensing*, Vol. 49, No. 3, pp. 357–364.

Goodman, J. W., 1976. Some fundamental properties of speckles. *J. Optical Society of America*, Vol. 66, No. 11, pp. 1145–1150.

Lee, J. S., 1980. Digital image enhancement and noise filtering by use of local statistics. *IEEE Trans. Pattern Anal. Mach. Intell.*, PAMI-2, No. 2, pp. 165–168.

———, 1981a. Refined filtering of image noise using local statistics. *Computer Graphics and Image Processing*, Vol. 15, pp. 380–389.

———, 1981b. Speckle analysis and smoothing of synthetic aperture radar images. *Computer Graphics and Image Processing*, Vol. 17, pp. 24–32.

———, 1983a. Digital image smoothing and the sigma filter. Computer Vision, Graphics, and Image Processing, Vol. 24, pp. 255–269.

———, 1983b. A simple speckle smoothing algorithm for synthetic aperture radar images. *IEEE Trans. Syst. Man. Cybern.*, Vol. SMC-13, No. 1, pp. 85–89.

McDonnell, M. J., 1981. Box-filtering techniques. *Computer Graphics and Image Processing*, Vol. 17, pp. 65–70.

Nagao, M., and T. Matsuyama, 1979. Edge-preserving smoothing. *Computer Graphics and Image Processing*, Vol. 9, pp. 394–407.

Pratt, W. K., 1978. *Digital Image Processing*, John Wiley and Sons, New York, 750 p.

Rosenfeld, A., and A. C. Kak, 1982. *Digital Picture Processing*, Vol. 1, Academic Press, New York, 435 p.

Schowengerdt, R. A., 1983. *Techniques for Image Processing and Classification in Remote Sensing*. Academic Press, New York, 249 p.

Seidman, J. B., 1972. Some practical applications of digital filtering in image processing. *Proceedings of the Symposium of Computer Image Processing and Recognition*, Vol. 2, Department of Electrical Engineering, University of Missouri, Colombia.

Soha, J. M., D. J. Lynn, J. J. Lorre, J. A. Mosher, N. N. Thayer, D. A. Elliott, W. D. Benton, and R. E. Dewar, 1975. IPL processing of the Mariner 10 images of Mercury. *Journal of Geophysical Research*, Vol. 80, No. 17, pp. 2394–2414.

Tom, V. T., 1985. Adaptive filter techniques for digital image enhancement. *Proceedings Society of Photo-Optical Instrumentation Engineers*, Vol. 528, pp. 29–42.

Torres, V., L. Wood, and N. Shah, 1988. Adaptive filtering for enhancing geological features. *Proceedings of the Sixth Thematic Conference on Remote Sensing for Exploration Geology*, Houston, Texas, pp. 691–697.

Wang, D., A. Vagnucci, and C. Li, 1981. Image enhancement by gradient inverse weighted smoothing scheme. *Computer Graphics and Image Processing*, Vol. 15, pp. 167–181.

(Received 13 June 1989; accepted 11 July 1989)