Image Smoothing by Local Use of Global Information

K. A. NARAYANAN AND AZRIEL ROSENFELD, FELLOW, IEEE

***Abstract*—There exists a class of iterative local image smoothing techniques in which a neighborhood of each pixel is examined, and the pixel is replaced by an average of a selected set of its neighbors chosen so as to make it likely that they belong to the same region as the pixel, e.g., on the basis of their resemblance to it. Thus these methods choose the neighbors to be averaged on the basis of information local to the neighborhood. A more powerful approach in which the neighbors are chosen based on global information derived from the histogram of the image is described. This approach attempts to choose neighbors that belong to the same histogram peak as the given pixel, but are more typical of that peak. Smoothing using this approach gives dramatically better results than smoothing based only on local information in cases where the noise in a region belongs to the same histogram peak as the region's mean.**

I. Introduction

There exists a class of iterative local image smoothing techniques in which a neighborhood of each pixel is examined, and the pixel is replaced by an average of a selected set of its neighbors, chosen so as to make it likely that they belong to the same region of the image as the pixel. For example, one can average the pixel with those k of its neighbors (for some fixed k) which are closest to it in gray level, or with those neighbors whose gray level gradient magnitudes are lower than that of the pixel itself [1]; or one can examine a set of subneighborhoods and use the average of the one that has least variable gray level [2], [3], that has the strongest contrast with its complement [4], or that is approximated by a polynomial that most closely fits the pixel itself [5].

All of those methods choose the neighbors that are averaged on the basis of information local to the neighborhood. This note describes a more powerful approach in which the neighbors are chosen based on global information derived from the histogram of the image. This approach attempts to choose neighbors that belong to the same histogram peak (i.e., to the same pixel subpopulation) as the given pixel but are more typical of that peak. As we shall see, smoothing using this approach gives dramatically better results than smoothing based only on local information, provided that the noise in a region does not belong to a different histogram peak than the region's mean. All of the methods described below used 3×3 neighborhoods.

II. Methods

The simplest of our histogram-based methods uses those neighbors whose probabilities, as estimated from the histogram, are higher than that of the given pixel. We will refer to this scheme as method I. A related idea was introduced in [4], namely, to average each pixel with those k of its neighbors, for some fixed k, that have the highest probabilities as estimated from the histogram. However this scheme did not work as well as method I.

If the neighbors of a pixel always belong to the same histogram peak as the given pixel or to a valley adjacent to that peak, then method I will tend to move the pixel higher up on that peak so that the peak will become sharper. (In fact, as we shall see, the peak will become a spike after a few iterations.) On the other hand, if neighbors can belong to other peaks and have higher probabilities than the pixel itself, method I will not work properly. For example, if the pixel belongs to a small peak and has neighbors belonging to a larger peak (e.g., a pixel on the border of a small region, adjacent to a large region), it may shift toward the large peak, causing the small region to shrink. This would be especially likely in the case where the two regions give rise to a peak and a shoulder rather than two peaks with a valley between them.

Method II is designed to overcome these potential problems by using a neighbor only if there is no significant concavity in the histogram between the pixel and the neighbor. (We also still require the neighbor to have higher probability than the pixel.) The presence of a concavity indicates that the pixel and the neighbor belong to two different peaks or to a peak and a shoulder, and we do not want such neighbors to influence each other.

To define method II more precisely, let Z and Z' be the gray levels of the pixel and the neighbor, let p and p' be their probabilities (proportional to their histogram bar heights), and let $s' = (p - p')/|Z' - Z| > 0$. Let Z'' be any gray level between Z and Z', let p'' be its probability, and let $s'' = (p'' - p)/|Z' - Z|$. If there exists a Z'' such that $s'' < s'/k$, where k is a parameter, we do not use the neighbor Z'. Evidently the existence of such a Z'' implies that the histogram has a concavity between Z and Z'. The higher k, the less likely that such a Z'' exists (the deeper the concavity must be) and the more likely we are to use the neighbor Z'.

In both methods I and II, it is desirable to smooth the histogram (at each iteration) before making decisions about the neighbors. This makes it less likely that the results will be influenced by fluctuations in the histogram due to quantization noise.

Method II assures, in principle, that a pixel will not get averaged with neighbors that do not belong to the same histogram peak. This should prevent small regions from shrinking and so is an improvement over method I. On the other hand, suppose that there are isolated noise pixels in a region that come from a different histogram peak (not the same one to which the region belongs); then such a pixel will be completely unaffected by method II, since all of its neighbors belong to a different peak than it does! Similarly, in the case of method I, suppose that an isolated noise pixel coming from a high peak is found in a region belonging to a low peak; then that pixel will be unaffected, and even worse, it will influence its neighbors since it is more probable than they are. Thus we see that methods I and II are reliable only if the noise in a region belongs to the same histogram peak as the region.

These observations imply that certain simple types of noise will not be removed by our methods. Perhaps the simplest case is that of salt-and-pepper noise in a binary image. Here the histogram consists of two spikes, and method II will have no effect at all since there is a deep concavity between the spikes. Method I, on the other hand, will cause the more probable color (black or white) to expand into the less probable color.

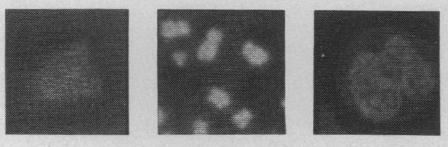
In spite of this limitation our methods seem to be extremely powerful in practice, as we shall see in the next section. In particular, method II has no effect on noise pixels that belong to a different peak than the surrounding region, but it still smooths the rest of the region so that its peak becomes a spike. It is then very easy to detect the noise pixels and remove them by simple methods, e.g., replace the pixel by the mean of its neighbors if (nearly) all of those neighbors differ from it by an above-threshold amount [6].

Manuscript received April 2, 1981. This work was supported by the National Science Foundation under Grant MCS-79-23422.

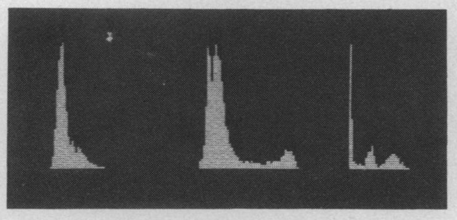
K. A. Narayanan is with the Computer Vision Laboratory, Computer Science Center, University of Maryland, College Park, MD 20742, on leave from ISRO Satellite Center, Bangalore, India.

A. Rosenfeld is with the Computer Vision Laboratory, Computer Science Center, University of Maryland, College Park, MD 20742.

0018-9472/81/1200-0826$00.75 ©1981 IEEE

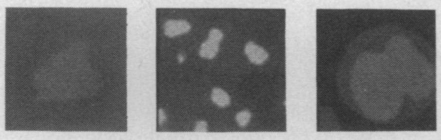


(a)

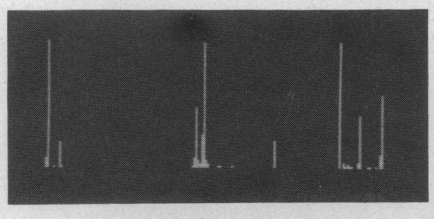


(b)

Fig. 1.

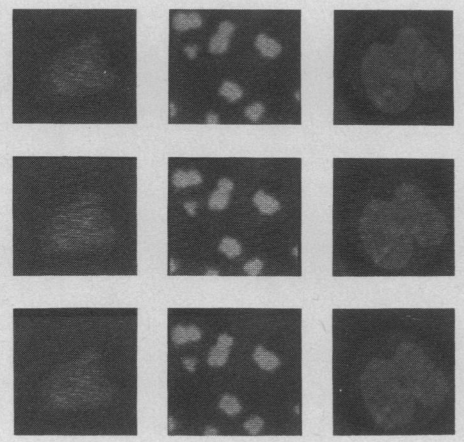


(a)

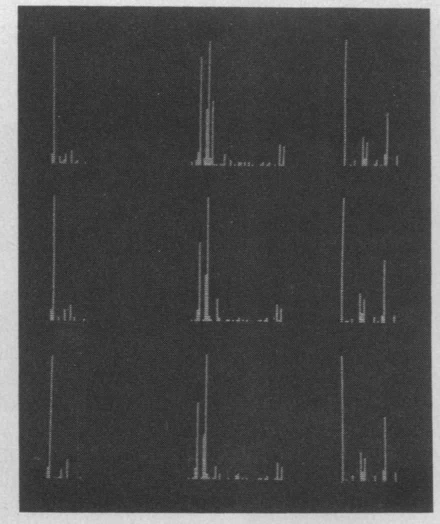


(b)

Fig. 2.



(a)



(b)

Fig. 3.

III. Experiments

The methods were first tested on the three images shown in Fig. 1(a); they are an infrared image of a tank, a portion of a chromosome spread, and a portion of a blood smear showing a white blood cell. The histograms of these images are shown in Fig. 1(b).

The results for these three images are shown in Fig. 2, using the following methods.

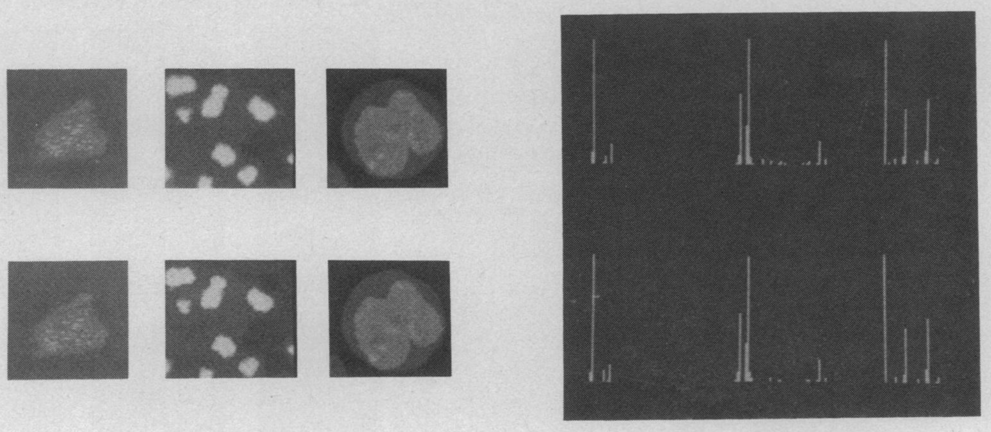
Figure	Method	k	Iterations of Histogram Smoothing
2	I	—	0
3	II	1, 2, 4	0
4	II	10, 100	0
5	II	1, 2, 4	1
6	II	10, 100	1
7	II	1, 2, 4	2
8	II	10, 100	2

In each case the result of six iterations of the method are shown. Histogram smoothing was done by taking a simple running average of three consecutive histogram bins. Part (a) of each figure shows the smoothed images, and part (b) shows their histograms. Note that the histograms have essentially become small sets of spikes, confirming that the smoothing was extremely effective. For comparison, Fig. 9 shows the results of six iterations of median filtering applied to the images; the histogram peaks have been sharpened but have not become spike-like.

The following comments may be made about the methods used based on these results. Method I yields spikes corresponding to the major peaks (or shoulders in the tank example) in the histogram, but it causes some shrinkage in small regions (see especially the chromosome example). Method II yields more spikes unless the histogram is smoothed, but this may not always be undesirable since it may reflect subpopulations of pixels within the regions. Both smoothing the histogram and increasing k tend to reduce the number of spikes (as should be expected), but small regions still do not shrink.

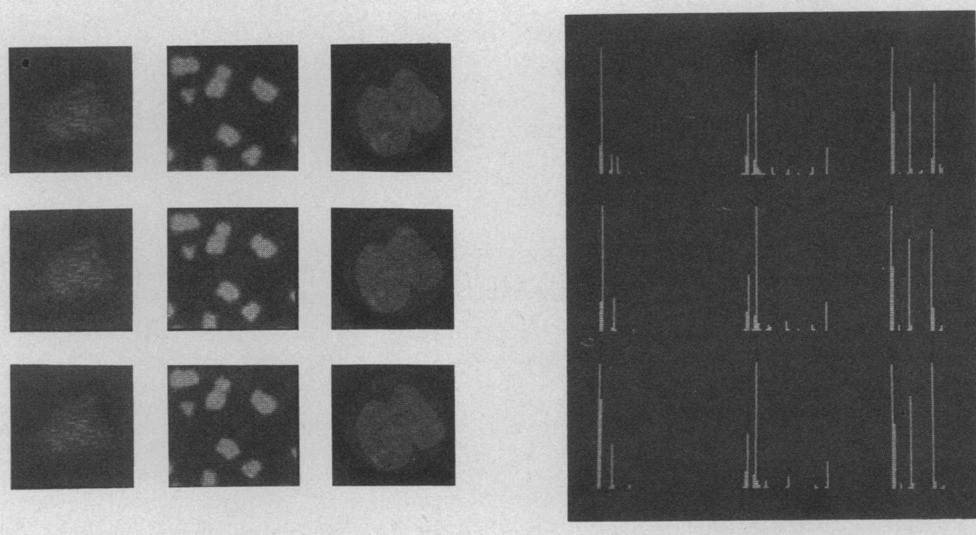
Fig. 10–14 show similar results for a portion of a Landsat image which was also used in [4]. The original image is shown in Fig. 10, while Figs. 11–14 show the following smoothing results.

Figure	Method	k	Iterations of Histogram Smoothing
11	I	—	0
12	II	1, 4, 10, 100	0
13	II	1, 4, 10, 100	1
14	II	1, 4, 10, 100	2



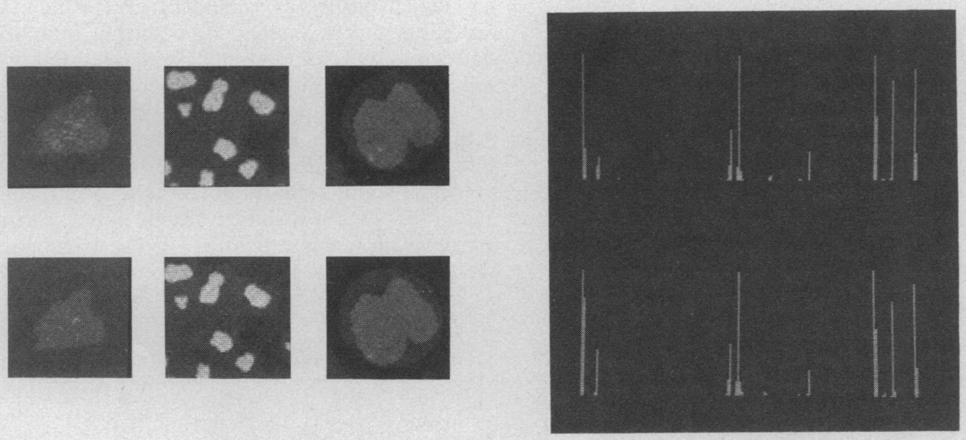
(a) (b)

Fig. 4.



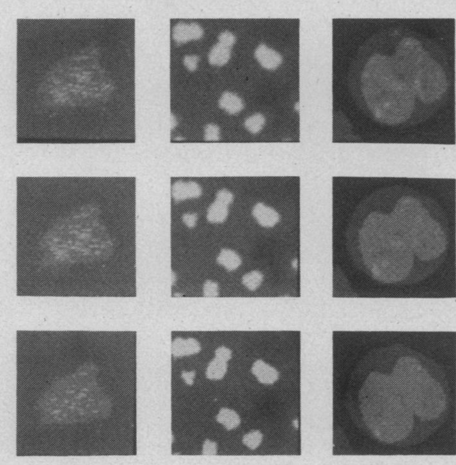
(a) (b)

Fig. 5.

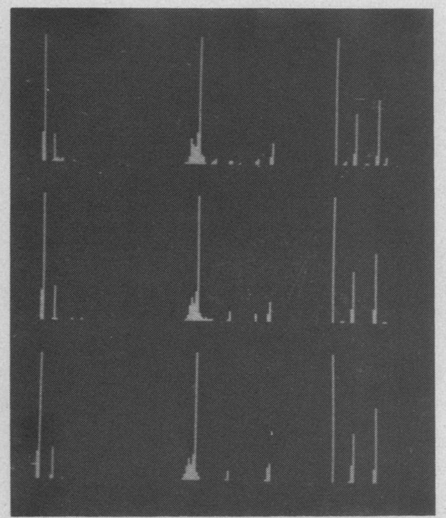


(a) (b)

Fig. 6.

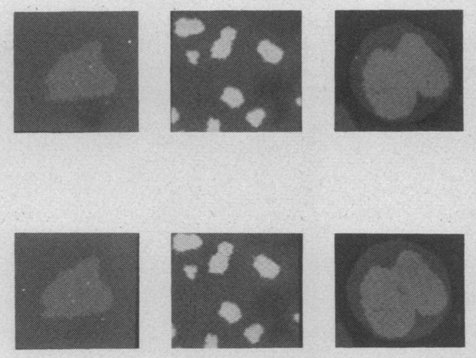


(a)

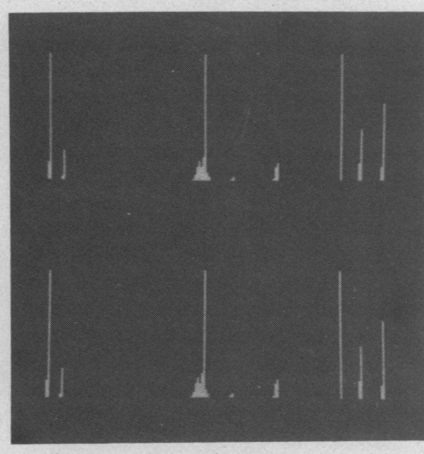


(b)

Fig. 7.

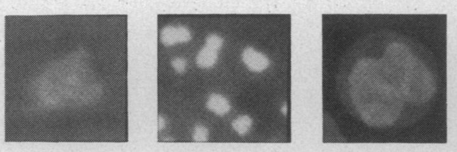


(a)

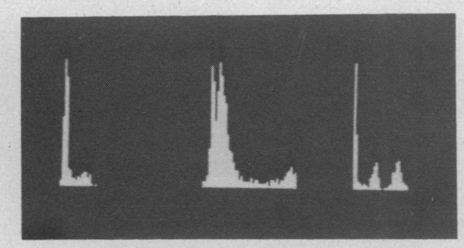


(b)

Fig. 8.



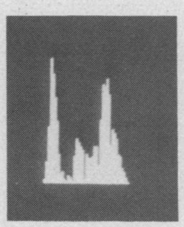
(a)



(b)

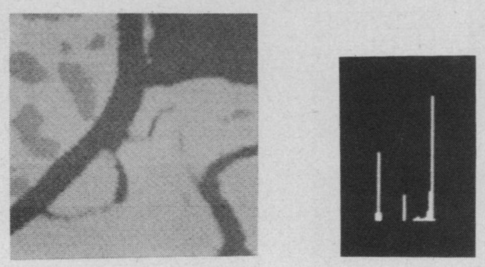
Fig. 9.

(a)



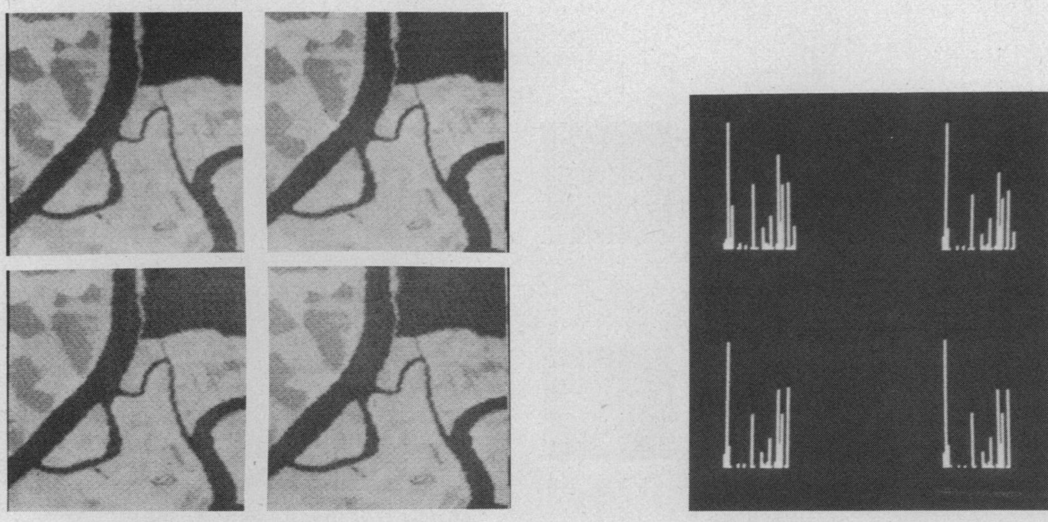
(b)

Fig. 10.



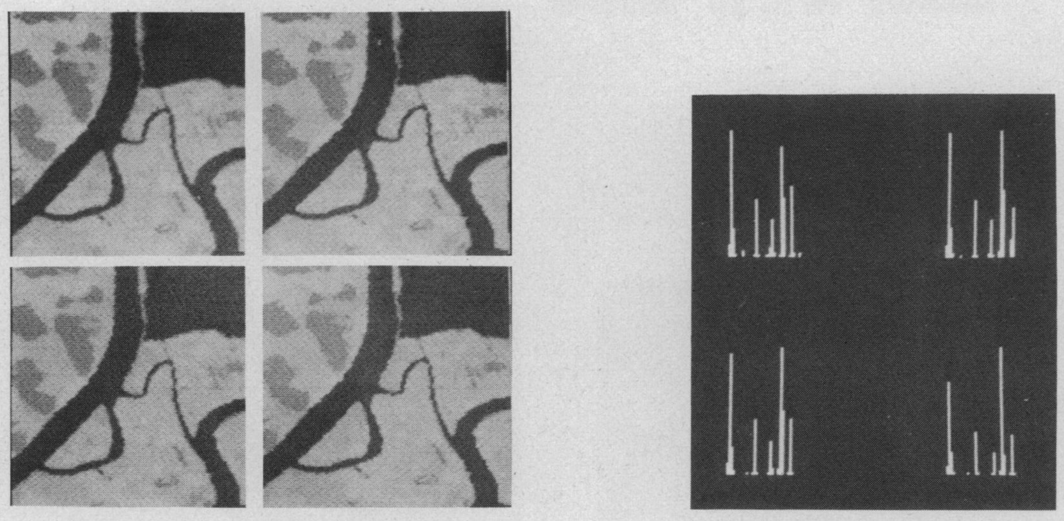
(a) (b)

Fig. 11.



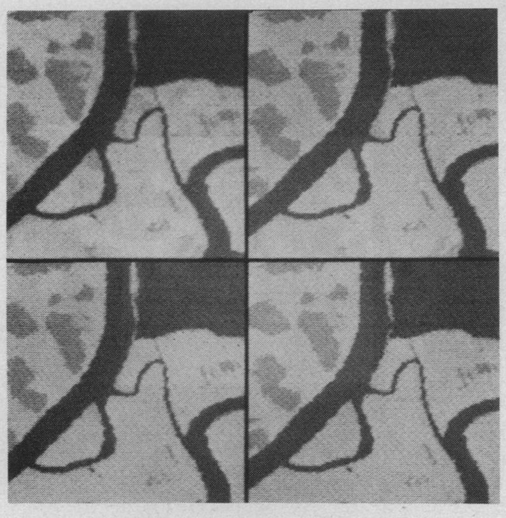
(a) (b)

Fig. 12.

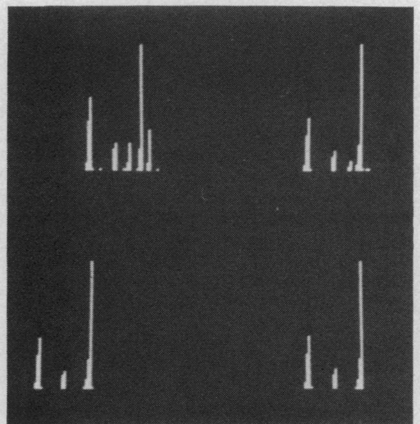


(a) (b)

Fig. 13.



(a)



(b)

Fig. 14.

The conclusions suggested by Figs. 2–8 are supported by these results as well; note especially the shrinkage of the thin dark regions under method I but not under method II even for high values of k. Evidently method II is preferable even though it has a somewhat greater computational cost.

IV. Concluding Remarks

This correspondence has investigated image smoothing by iterative local averaging of each pixel with those of its neighbors that have higher histogram values and belong to the same histogram peak. Even though only a few iterations were performed using a small neighborhood (3×3), the smoothing was extremely effective on a variety of images, turning their histograms into small sets of spikes. The approach works only on noise that belongs to the same histogram peak as the region it lies in; noise belonging to other peaks will be unaffected. However once the rest of the region has become smooth, such noise can easily be detected and deleted by conventional methods.

The computational cost of this approach is not very high. By examining the histogram we can define, for each gray level, the set of gray levels with which averaging is permitted; this lookup table can then be used when each pixel is processed. In any case the power of the histogram-based approach makes it a serious contender for use in practical image smoothing applications and illustrates the advantages of introducing global information into local image processing operations.

Acknowledgment

The help of Sherry Palmer in preparing this correspondence is gratefully acknowledged.

References

[1] L. S. Davis and A. Rosenfeld, "Noise cleaning by iterated local averaging," *IEEE Trans. Syst. Man, Cybern.*, 1978, vol. SMC-8, pp. 705–710.

[2] F. Tomita and S. Tsuji, Extraction of multiple regions by smoothing in selected neighborhoods, *IEEE Trans. Syst., Man, Cybern.*, 1977, vol. SMC-7, pp. 107–109.

[3] M. Nagao and T. Matsuyama, "Edge preserving smoothing," *Computer Graphics and Image Processing*, vol. 9, 1979, pp. 394–407.

[4] A. Scher, F. R. D. Velasco, and A. Rosenfeld, "Some new image smoothing techniques," *IEEE Trans. Syst., Man, Cybern.*, 1980, vol. SMC-10, pp. 153–158.

[5] R. M. Haralick and L. Watson, "A facet model for image data," *Proc. IEEE Conf. on Pattern Recomposition and Image Processing*, vol. 1979, pp. 489–497.

[6] A. Rosenfeld and A. C. Kak, *Digital Picture Processing.* New York: Academic, 1976, Sec. 6.5.

A Note on Invariant Moments in Image Processing

TIEN C. HSIA, SENIOR MEMBER, IEEE

***Abstract*—A number of observations on the application of moments to the classification and matching of digital images are presented. Supported by experimental data, comments were made on the effects of contrast change, correlation, and ranking of invariant moments.**

Introduction

The method of invariant moments is a very useful tool for extracting features from two-dimensional images which are invariant with respect to the images position, size, and orientation. Applications of invariant moments method to character recognition, image classification and scene matching have been reported with varying degrees of success [1]–[6]. Examination of these published results showed that further studies are needed in order to enhance our understanding and effective use of this method. The objective of this correspondence is to present a number of observations based on some experimental data obtained from digital images.

There are a number of invariant moments which can be computed from a given two-dimensional image intensity function $f(x, y)$. Following are the first seven moments which are commonly used for scene matching [6]

$$\phi(1) = \eta_{20} + \eta_{02}$$

$$\phi(2) = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2$$

$$\phi(3) = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2$$

$$\phi(4) = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2$$

$$\phi(5) = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})\left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\right] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})\left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right]$$

$$\phi(6) = (\eta_{20} - \eta_{02})\left[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})$$

$$\phi(7) = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})\left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\right] - (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})\left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right].$$

Manuscript received December 18, 1980; revised July 23, 1981.
The author is with the Department of Electrical and Computer Engineering, University of California, Davis, CA 95616.