COMPUTER VISION, GRAPHICS, AND IMAGE PROCESSING 46, 82–95 (1989)

A New Method for Image Segmentation*

S. D. YANOWITZ AND A. M. BRUCKSTEIN

Faculty of Electrical Engineering, Technion, IIT, 32000 Haifa, Israel

Received February 4, 1988; revised September 28, 1988

In applications involving visual inspection, it is often required to separate objects from background, in conditions of poor and nonuniform illumination. In such cases one has to rely on adaptive methods that learn the illumination from the given images and base the object/background decision on this information. We here present a new method for image segmentation via adaptive thresholding. The threshold surface is determined by interpolating the image gray levels at points where the gradient is high, indicating probable object edges. Several methods of data interpolation to levels given at scattered points in the image plane are discussed. One method is tested on several examples and the segmentation results are compared to previously proposed adaptive thresholding algorithms. © 1989 Academic Press, Inc.

1. INTRODUCTION

In a variety of applications involving visual inspection, it is required to separate objects from background in an image taken under conditions of poor and nonuniform illumination. Given that the objects in an image appear lighter (or darker) than the background, one may attempt to segment the image (i.e., separate objects from background) by means of thresholding. A certain threshold level is chosen, and one assigns the label "1" to each image pixel whose gray level is higher than the threshold, and a "0" label to all pixels whose gray level is lower. Threshold selection is usually made based on the information contained in the gray-level histogram of the image. Bimodality of the histogram often indicates that the gray levels corresponding to one mode are populated with pixels that form the objects within the image, and the gray levels corresponding to the other mode are populated with the rest of the pixels which form the background. Thus, the threshold level is selected to best separate the modes of the histogram, see [1–5]. Rosenfeld and Panda [4] suggested that it would be wiser, when measuring the gray-level histogram, to only count those pixels corresponding to low gray-level gradients in the image, since in this manner we are likely to take into account only pixels belonging to either the objects or the background, disregarding the pixels belonging to boundaries where the gray level varies rapidly. This should yield a bimodal histogram with better defined modes, the valley between the modes being a proper threshold level. Conversely, if we count only those pixels having high gradient values it is likely that we take into account only pixels belonging to boundaries, and these should yield a well-defined unimodal histogram, the peak value of which is a proper constant threshold level. These ideas can be merged into measuring a 2-dimensional histogram or "scatter diagram" having gray level and gradient as its coordinates and using clusters of points, which replace the modes of the histogram, to select a good threshold.

*This research was supported in part by the Foundation for Research in Electronics, Computers and Communications, administered by the Israel Academy of Sciences and Humanities.

0734-189X/89 $3.00

A fixed threshold level, no matter how well chosen, cannot be appropriate in cases of uneven background and poor illumination. In such cases, although locally the objects will still be lighter or darker than the background, the histogram may not be bimodal at all, or it might be bimodal by chance only. In any case, a constant threshold level induced by the gray-level histogram will fail to properly separate the objects from the background. We need here a threshold level that varies over different image regions so as to fit the spatially changing background and lighting conditions. In other words we need a threshold *surface*.

Chow and Kaneko [6] have suggested a method for obtaining a threshold surface, and this method was further studied by Nakagawa and Rosenfeld [7]. The idea underlying the Chow–Kaneko procedure is to use local information on gray-level distributions to create an adaptive threshold surface. They suggested to divide the image into nonoverlapping cells of equal area forming a regular grid and to determine the (sub)histograms of the gray levels of pixels in each cell. Then the subhistograms that are judged to be bimodal are used to determine local threshold values for the corresponding cell centers, and the local thresholds are interpolated over the entire image to yield a threshold surface. This method is certainly an improvement over fixed thresholding since it utilizes some local information. However, the local information is implicitly blurred to the dimensions of the grid cell. It is clear that we cannot reduce the size of the cell too much, trying to focus on local properties, since this would reduce the number of pixels participating in the subhistogram to an extent that would yield meaningless statistics. Note that the grid imposed on the image is not in any coherence with the image contents. The threshold values determined within a cell are set at arbitrary locations, instead of being placed in truly meaningful positions. Furthermore, serious errors occur if, due to noise and bad lighting conditions, grid cells placed entirely on object areas or entirely on background areas, by chance yield subhistograms that are judged to be bimodal. Note also that the method utilizes gray level distributions alone without considering extra information such as the gradient. In experiments, the Chow–Kaneko methods yields good results, however, in some test cases it fails to give proper segmentations. In fact, in one of the experiments described in [7] it gave poorer results than the segmentation obtained with fixed thresholding.

A more refined approach is to use information additional to that of the gray-level distribution, usually gradient or edge information, to help the process of separation between objects and background. The so-called "superslice" method, described in [8], employs additional edge information. The method assumes that a single fixed threshold level may not be enough to properly segment the image, but that several fixed levels, possibly a different level for every individual object will be necessary. The image is sliced according to various threshold levels, and subsequently an object validation process is used. In this process the boundary of each object in the segmented image is scanned and, from the original image, the average gradient level along the boundary is computed. If for a particular object, the averaged edge values do not give enough support for its existence, that object is considered improperly segmented and another threshold is tested. The process is continued until for each object a threshold level is obtained, and in the segmented image each object is properly validated by the gradients along its boundary. The validation process thus provides feedback for setting the local threshold levels, and effectively removes stains that cannot be validated for any threshold level.

Although this method too is an improvement over fixed thresholding, a logical next step would be to first segment the image using an adaptive variable-threshold surface, and then use a similar validation process to correct the result. This would solve the problem of objects that cannot be properly segmented with a fixed threshold due to illumination variations over their surface.

The purpose of this paper is to present a new method for finding a threshold surface which involves the ideas employed in the above methods but attempts to overcome some of their disadvantages. The method uses the gradient map of the image to point at well-defined portions of object boundaries in it. Both the location and gray levels at these boundary points make them good choices for local thresholds. These point values are then interpolated, yielding the threshold surface. A method for fitting a surface to this set of points which are scattered in a manner unknown in advance becomes necessary and we discuss several possible choices describing in detail the implementation of one of them. Finally, we apply two versions of the Chow and Kaneko algorithm and ours to a few images and compare the results.

We note that there also exist other methods for segmentation of a different nature such as region growing [9], split and merge [10], and probabilistic relaxation [11]. Reference [12] surveys these and the various available thresholding methods.

2. THE NEW ADAPTIVE THRESHOLDING METHOD

The fixed thresholding approach, as well as the Chow–Kaneko method are based on gray-level distributions only. Although the adaptive method of threshold selection due to Chow and Kaneko, and the superslice method too, do take into account spatial variations due to uneven background and illumination conditions, the thresholds chosen by them are still not completely adapted to the image content. We thereby lose important local information contained in the image space. This discussion indicates that it would be better to combine gray level and gradient information in a more direct way in the original image plane. We attempt to achieve this in the method described below. First note that for clean images with sharply defined object boundaries, the gradient alone may suffice for segmentation since it outlines the contours that can later be filled to obtain the objects. However, noise and poor illumination may break contours, leading to the loss of entire objects. For partially avoiding such problems, contour-following algorithms have been proposed (see [1]). The method we propose is to use combined edge and gray-level information in a natural way to obtain a good threshold surface and then to use a validation process to clean the image from stains that are due to random illumination variations.

Given an image to be segmented, we first derive a gradient magnitude image from the original image. We then threshold the gradient and, for the regions where the gradient is above the chosen threshold, we apply a process determining thinned paths of local gradient maxima (that may degenerate into isolated points, in some cases). These, hopefully, robustly determine locations of maximal slope in the boundary areas separating the objects from the background, and their gray levels are good local threshold values. We therefore use the points of maximal gradient as "pointers" to positions at which we should sample the gray levels of the original image. Interpolating the sampled levels over the entire image yields the desired threshold surface. This process is illustrated schematically in Fig. 1. The steps of the

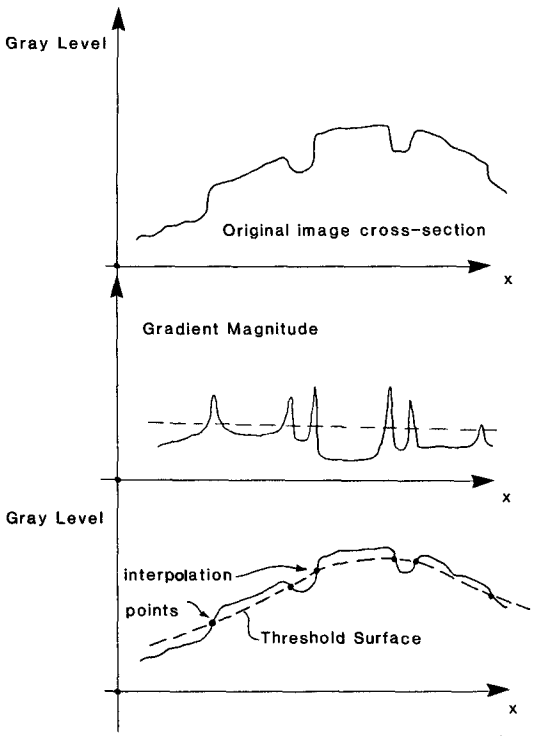

FIG. 1. Description of the process of determining the adaptive threshold surface: (top) cross section in the original image, showing objects on an uneven background, (middle) cross section of the gradient magnitude image, (bottom) peak points in the gradient point at interpolation values, that determine the threshold surface.

algorithm are:

Step 1. Smooth the image replacing every pixel by the average gray-level values of some small neighborhood of it. This serves a few purposes. It is easier to separate the objects from the background when the image is smoother, see [1, 2, 5]. Averaging also makes the samples we take less vulnerable to errors both in magnitude and location, as slopes become more moderate.

Step 2. Derive the gray-level gradient magnitude image from the smoothed original image.

Step 3. Apply thresholding and a local-maxima-directed thinning processes to the gradient, obtaining paths and points in the image plane having local gradient maxima. These point at pixels at which the original image gray levels are good candidates for local threshold.

Step 4. Sample the smoothed image at the places at which the maximal gradient-mask points. The gray-level values of the so-determined set of points are to be interpolated over the image. The interpolation process that is described in the sequel requires that all values on the border of the image also be supplied. We could either set the values at the border to the original image gray levels or supply somewhat higher values, making sure that the interpolation surface will be above the image surface near the border, so that no false objects will be detected there. However, it may happen that objects close to the image border are cut by the image

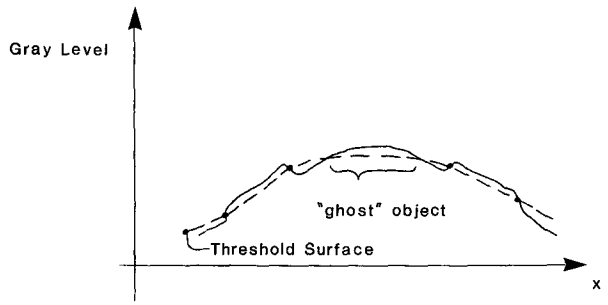

FIG. 2. Cross section in an image that illustrates a possible problem where thresholding yields a "ghost" image. The validation process removes such unwanted objects.

frame. Thus, supplying high border values would lead to total or partial loss of those objects. The solution chosen was to derive a one-dimensional gradient along the border line (like we do in two dimensions), seek for object boundaries there, and if such boundaries are found, sample the boundary values and 1-dimensionally and linearly interpolate them along the frame line. Thus the border values are set as a result of a 1-dimensional segmentation process.

Step 5. Interpolate the sampled gray levels over the image. Note that the manner in which the sample points are scattered is unknown in advance. The interpolation surface may be obtained in several ways. We could use procedures that are simple but can only approximate the gray levels at the given points, or we may insist that the interpolation surface pass exactly through the data. The second choice is better but, as discussed below, algorithms for exact interpolation require more storage space. We have chosen to implement the latter approach and will describe such an interpolation process later.

Step 6. Using the obtained threshold surface, segment the image.

Step 7. Due to uneven illumination and noise, the threshold surface might intersect background or object regions as illustrated in Fig. 2, yielding "ghost" objects and stains in the segmented image. These stains, however, can be removed by a validation process, since the gradient values along their boundaries should be low. The validation process can in fact be incorporated in all methods of segmentation. As discussed in the Introduction, such a procedure was first proposed by Milgram, Rosenfeld, *et al.*, [8], in the framework of the superslice algorithm. The validation process first scans the segmented image, labeling different connected components as different objects. Note that labeling should be done separately for black connected components and for white ones, since the "ghost" objects can take the form of either black stains on white background or vice versa. After labeling, the process scans the boundaries of all labeled objects and looks up the corresponding edge values in the gradient map derived earlier. If for a particular object the average of its edge values does not exceed a certain threshold, the process eliminates it.

3. INTERPOLATION WITH POTENTIAL SURFACES

There are several approaches to our 2-dimensional interpolation problem. The first group of methods uses parametrized classes of (usually radially symmetric) base

functions and sets their parameters so as to approximate the given data points with minimal approximation error, see [13, 14]. This approach considerably reduces the storage space required for the interpolation surface since we may store the parameters of a few base functions and compute the surface values by a formula. The other type of approach is to require the interpolation surface to pass exactly through the data points. Since a typical image has more than a quarter of a million points, having a surface pass exactly through a set of arbitrary points of that image, implies that the interpolation surface will have to be stored at all points and thus the memory required for the solution is that of an entire image.

Grimson [15] proposed an approach to solve the exact interpolation problem as a minimization of some cost function depending on the local derivatives of the image, such as the gradient or the Laplacian. What we shall use in the sequel is a direct procedure that, instead of minimizing the Laplacian of the surface $P(x, y)$ nulls it. Thus we exactly solve the Laplace equation (1):

$$\frac{\partial^2 P(x, y)}{\partial x^2} + \frac{\partial^2 P(x, y)}{\partial y^2} = 0. \tag{1}$$

Surfaces $P(x, y)$ satisfying Eq. (1) are called *potential surfaces*, for obvious reasons. There are several good reasons for interpolating with potential surfaces. The physical meaning of the Laplace equation is that the divergence of the gradient of the potential surface vanishes everywhere. The fact that the gradient has zero curl states that the surface should be smooth enough for our purposes.

An alternative exact interpolation procedure would be to compute the value $P(x, y)$ of each unknown point as a weighted average of all given values P_i using Eq. (2):

$$P(x, y) = \sum_i \frac{W_i(x, y) P_i}{\sum_j W_j(x, y)}, \tag{2}$$

where $W_i(x, y) = 1/\Phi\{d[(x, y), (x_i, y_i)]\}$ such that $\Phi\{d\}$ is a monotonically increasing function of the distance d between the data points (x_i, y_i), where the interpolation function should be P_i, and we also have $\Phi\{0\} = 0$. This method ensures that points P_i closer to (x, y) have much higher weights and clearly, the limiting values when approaching the data point locations (x_i, y_i) are, as required, P_i. This and related approaches are described by Franke [16]. It is easy to show that for a $N \times N$ image, having M interpolation values (where M is usually much larger than N), this process requires about M scans of the image, with $4M(N \times N - M)$ multiplications and $M(N \times N - M)$ square root computations. On the other hand, the successive-over-relation method for finding the potential surface, which is described below, requires about N scans, with $N(N \times N - M)$ multiplications. If the image has 512×512 pixels with 5,000 interpolation points, the latter method would be more than 40 times faster.

Another potentially good, and altogether different solution would be to use *minimal surfaces* which are defined as the exact interpolation surfaces that have the minimal area. Such surfaces are usually described as the shape of soap bubbles and are important in architecture, for structural designs, not only because they require less material to construct, but also because the tensile forces prevailing upon them are uniform and minimal in the Minmax sense, see [17, 18]. The minimal surface

equation, however, is the following nonlinear partial differential equation,

$$\frac{\partial^2 P}{\partial x^2}\left[1+\left(\frac{\partial P}{\partial y}\right)^2\right]+\frac{\partial^2 P}{\partial y^2}\left[1+\left(\frac{\partial P}{\partial x}\right)^2\right]-2\frac{\partial P}{\partial x}\frac{\partial P}{\partial y}\frac{\partial^2 P}{\partial x\,\partial y}=0. \tag{3}$$

and it is much more complicated to solve numerically. A solution of the above equation for special cases is given by Ishii [19]. Do Carmo [17] pointed out an interesting relationship between potential and minimal surfaces. It is easily seen that if the surface is isothermal, i.e., satisfies Eqs. (4) and (5),

$$\left(\frac{\partial P}{\partial y}\right)^2=\left(\frac{\partial P}{\partial x}\right)^2, \tag{4}$$

$$\frac{\partial P}{\partial x}\frac{\partial P}{\partial y}\frac{\partial^2 P}{\partial x\,\partial y}=0, \tag{5}$$

then it is minimal only if it also satisfies the Laplace equation; i.e., it is a potential surface.

We have chosen to implement in our examples an exact interpolation procedure based on potential surfaces. We solved the Laplace equation using Southwell's *successive over-relaxation method* described in [20]. We start with a surface having the interpolation points set at their correct values, including the boundary, but with arbitrary (say, 0) values everywhere else. Scanning the surface sequentially we compute the discrete Laplace operation (6) for every pixel,

$$\begin{aligned}\Delta P(x, y) &= P(x, y+1)+P(x, y-1)\\ &\quad +P(x-1, y)+P(x+1, y)-4P(x, y)\\ &= R(x, y).\end{aligned} \tag{6}$$

Since at first the surface does not satisfy the equation, the operator will give a residual value $R(x, y)$ other than zero for almost every pixel. We then correct the value of the pixel as follows

$$P_n(x, y)=P_{n-1}(x, y)+\frac{\beta\cdot R(x, y)}{4}. \tag{7}$$

In order to make the residua vanish for the particular pixel, we have to make β equal 1. However, Southwell found that choosing $1<\beta<2$, will make the convergence faster.

The interpolation points, already having correct values, are held fixed during the iterative process described above. These are the boundary conditions that determine the values of all other points. The iterative process is stopped when the maximal residuum in one iteration no longer exceeds some desired small value. Convergence strongly depends on the number and location of the boundary points. Practically, it took a Vax-750 computer about one hour to solve the Laplace equation for a 512×512 image. It also requires four times the storage space required for one image, since for convergence purposes we have the convert the surface from byte form to floating point form. In order to reduce both storage space and CPU time, we also determined approximate solutions on a diluted grid, replacing every 2×2 pixel window in the sampled image by the maximal value of that window (since most values are zero), afterwards interpolating the resulting surface back to its original size. This reduced the time to convergence to about five minutes, the

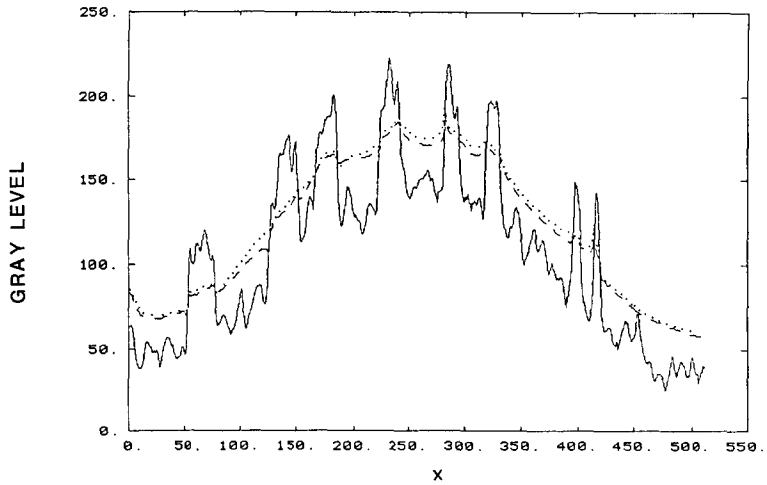

FIG. 3. Comparison of cross sections through threshold surfaces obtained by relaxation on the original grid and on a diluted grid, for the segmentation of the image presented in Fig. 5a.

approximate surface being very close to the real one in a particular test case, as seen in Fig. 3 (which presents a slice through the image and the interpolated adaptive threshold surfaces for the example of Fig. 5a). Since results were so close, further dilution of the grid may be recommended to reduce computation time.

4. EXPERIMENTAL RESULTS AND CONCLUSIONS

To test the proposed segmentation algorithm three images were constructed, as shown in Fig. 4. These images were spoiled by adding narrow-band noise that has slowly varying oscillations, white noise, and uneven background illumination, as shown in Fig. 5. The resulting images differ in the quality of edges, Fig. 5a possessing the best defined edges and Fig. 5c, the worst. In example 5b the quality of the edges greatly varies over the image. Figure 3 is a cross section through Fig. 5a and gives an idea about the amplitude of gray-level jumps at object boundaries in the best of the cases studied, as well as about the noise level and the type of nonuniform background that was added to the original images.

Three algorithms, the algorithm proposed above, the Chow–Kaneko procedure described in detail in [6], and also an improved version of the Chow–Kaneko algorithm due to Rosenfeld and Nakagawa [7], were then used to obtain threshold surfaces for these images. We note that the algorithm of [7], is the same as the original Chow–Kaneko method, but uses a more robust method for detecting the bimodality of partial histograms and for determining the corresponding local thresholds.

Figure 6 shows the threshold surfaces obtained for image 5a. Figures 7 and 8 show the resulting segmentations before and after validation for the same image. Validation levels (i.e., the threshold for the average gradient over the boundary) applied to all three images were equal.

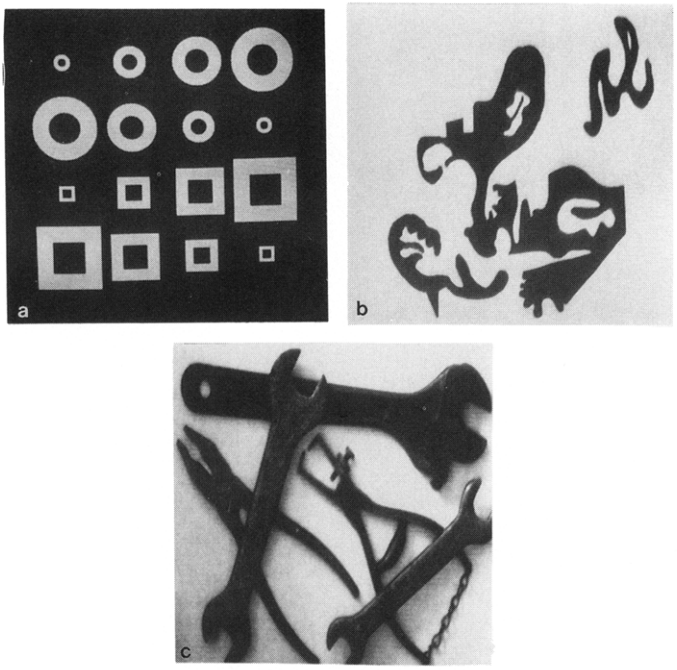

FIG. 4. The original images.

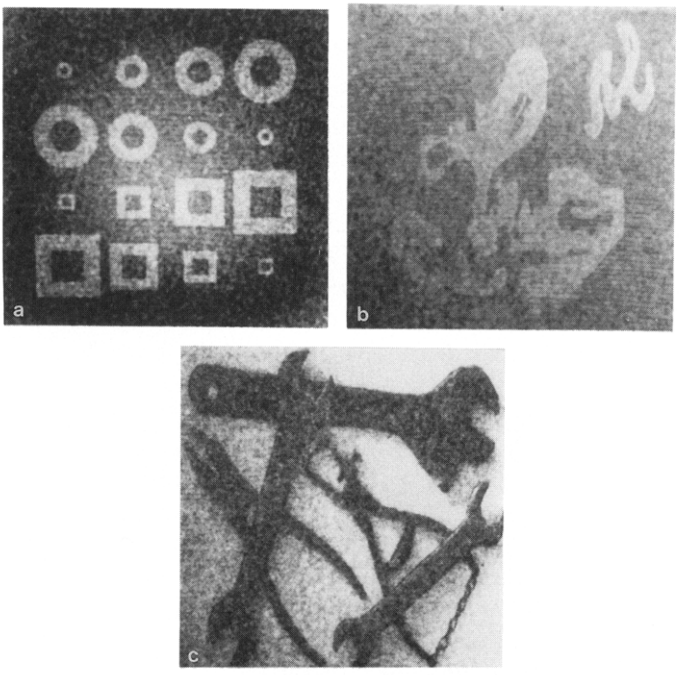

FIG. 5. The spoiled images used in testing the segmentation processes. Uneven background and noise was added to the original images of Fig. 4.

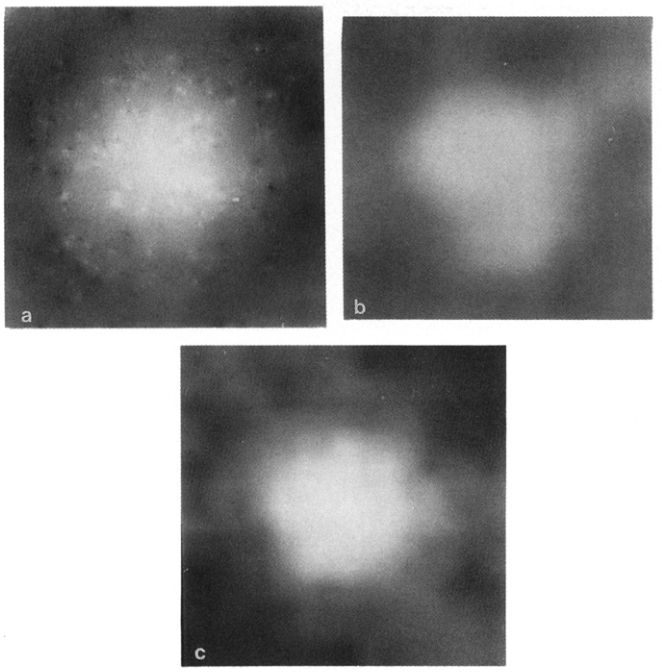

FIG. 6. Threshold surfaces for Fig. 5a: (a) using the proposed method with potential surface interpolation, (b) using the Chow–Kaneko process, and (c) using the modified algorithm of Rosenfeld and Nakagawa. (Same order of presentation is kept in all the figures that follow.)

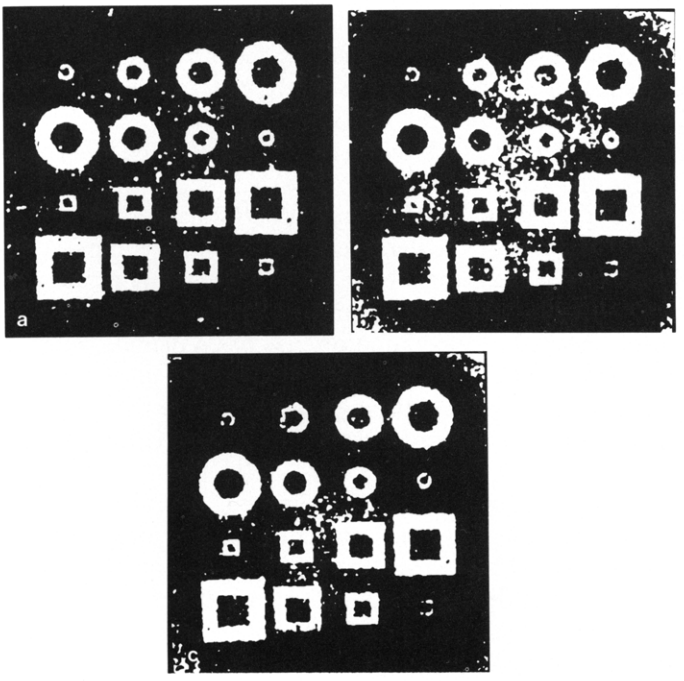

FIG. 7. Segmentation by adaptive thresholding of Fig. 5a, without validation.

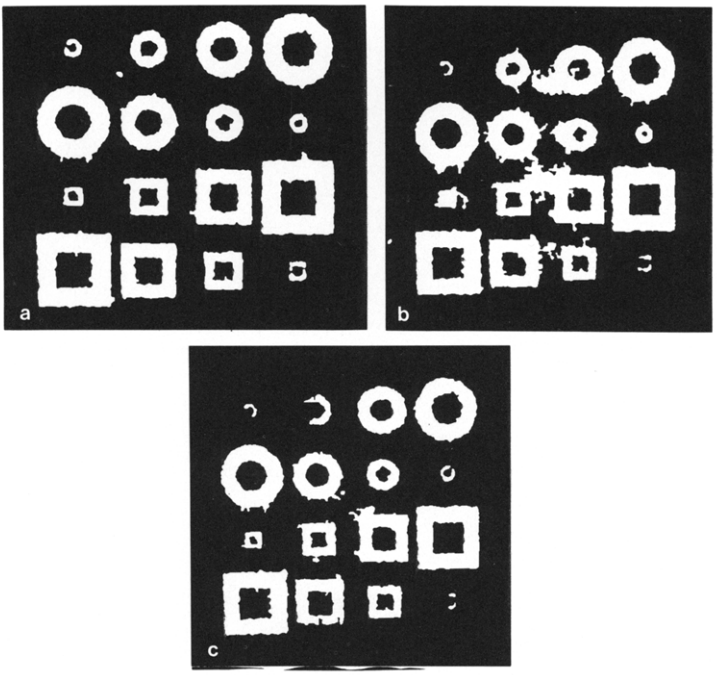

FIG. 8. Segmentation by adaptive thresholding of Fig. 5a, after validation.

It is seen that the proposed algorithm gave better results than the other two, the Rosenfeld–Nakagawa algorithm giving better results than the original Chow–Kaneko procedure.

Figures 9, 10, 11, and 12 show the segmentations of the other two images. Again, the proposed algorithm appears to give slightly better results than the other two. This time, however, the Rosenfeld–Nakagawa algorithm, which has more strict detection criteria than Chow and Kaneko, failed to detect the faint object at bottom left in the Fig. 5b, as seen in Fig. 10.

The results of Fig. 12 show almost identical performance of all three algorithms. This fact indicates the possible existence of a "performance improvement threshold effect" for the proposed algorithm. Note that the new algorithm depends on the correct detection of edges in the original image. Therefore when edges are, at least in part, well defined, the new segmentation process has a definite potential to outperform other adaptive thresholding methods. Using smart edge detection operators, and incorporating some contour following ideas, may further extend the range of images for which this new procedure will consistently outperform the other adaptive segmentation methods.

The human performance in segmenting an image is clearly the one that we should try to match in automatic segmentation procedures. Therefore a good way of testing our algorithms would be by comparing the segmentations obtained with it to human performance. We feel that the proposed algorithm matches the human performance quite well, as judged by the example of Fig. 5b, the one where a human subject cannot rely on any external information or regularity in the image to help in object-background decisions.

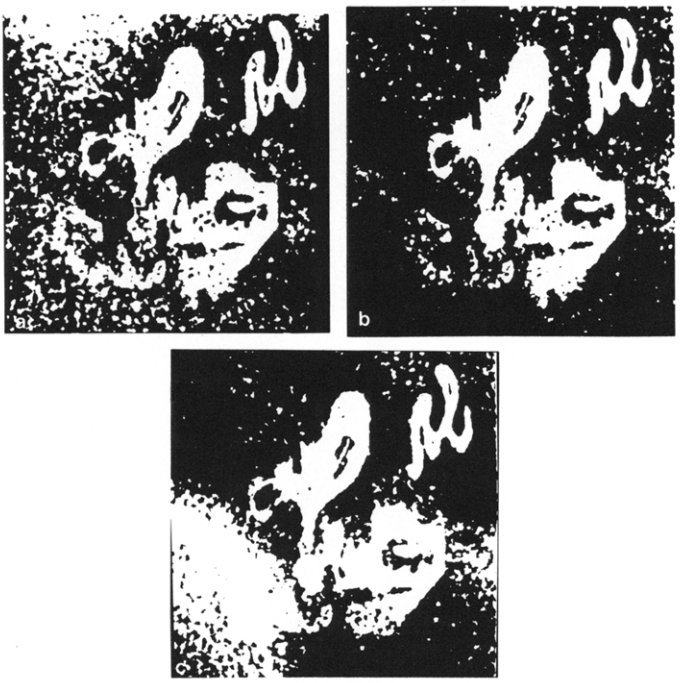

FIG. 9. Segmentation by adaptive thresholding of Fig. 5b, without validation.

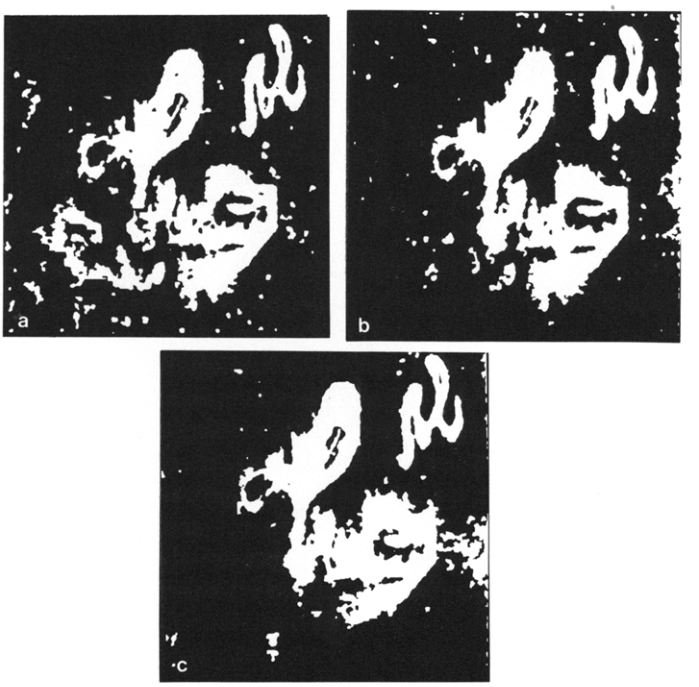

FIG. 10. Segmentation by adaptive thresholding of Fig. 5b, after validation.

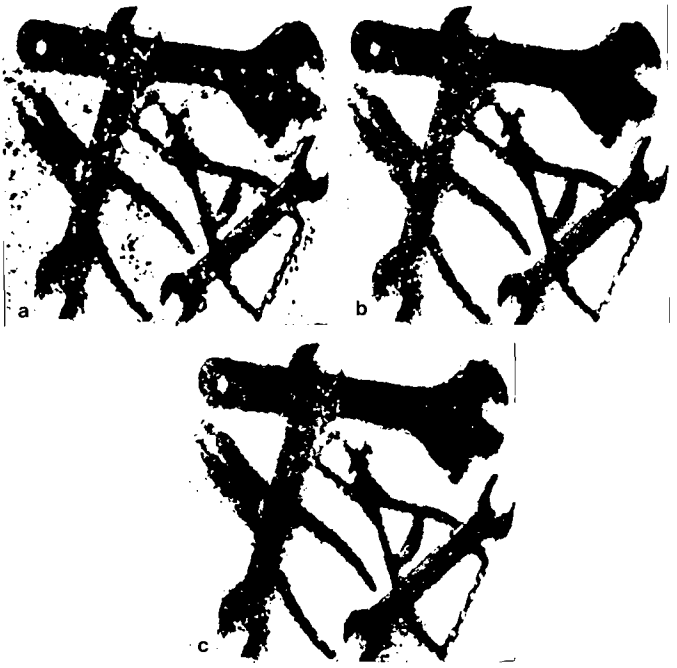

FIG. 11. Segmentation by adaptive thresholding of Fig. 5c, without validation.

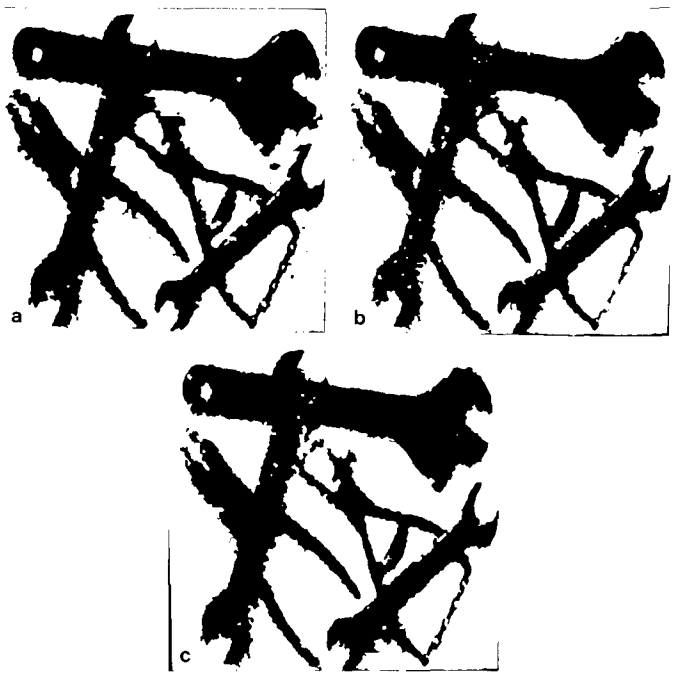

FIG. 12. Segmentation by adaptive thresholding of Fig. 5c, after validation.

ACKNOWLEDGMENTS

We thank Mr. Arkadi Gluchovski for his help in preparing the test images and in running some of the classical segmentation algorithms on these images.

REFERENCES

1. A. Rosenfeld and A. C. Kak, *Digital Picture Processing*, Academic Press, New York, 1982.
2. L. S. Davis, A. Rosenfeld, and J. S. Weszka, Region extraction by averaging and thresholding, *IEEE Trans. Systems Man Cybern.* **SMC-5**, 1975, 383–388.
3. J. S. Weszka, A survey of threshold selection techniques, *Comput. Graphics Image Process.* **7**, 1978, 259–265.
4. D. P. Panda and A. Rosenfeld, Image segmentation by pixel classification in (gray level, edge value) space, *IEEE Trans. Comput.* **C-27**, 1978, 875–879.
5. J. S. Weszka and A. Rosenfeld, Hystogram modification for threshold selection, *IEEE Trans. Systems Man Cybern.* **SMC-9** 1979, 38–52.
6. C. K. Chow and T. Kaneko, Automatic boundary detection of the left-ventricle from cineangiograms, *Comput. Biomed. Res.* **5**, 1972, 388–410.
7. Y. Nakagawa and A. Rosenfeld, Some experiments on variable thresholding, *Pattern Recognit.* **11**, 1979, 191–204.
8. D. L. Milgram, A. Rosenfeld, T. Willet, and G. Tisdale, *Algorithms and Hardware Technology for Image Recognition*, Final Report to U.S. Army Night Vision Laboratory, 1978.
9. S. Zuker, Region growing: Childhood and adolescence, *Comput. Graphics Image Process.* **5**, 1976, 382–399.
10. P. C. Chen and T. Pavlidis, Segmentation by texture using a co-occurrence matrix and a split-and-merge algorithm, *Comput. Graphics Image Process.* **10**, 1979, 172–182.
11. A. J. Dankner and A. Rosenfeld, Blob detection by relaxation, *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-3**, 1981, 79–82.
12. R. M. Haralick and L. Shapiro, Survey on image segmentation techniques, *Comput. Vision Graphics Image Process.* **29**, 1985, 100–132.
13. Y. Baram and M. Margalit, Surface fitting by pseudo-potential functions, *IEEE Trans. Geosci. Remote Sensing* **GE-22**, 1984, 455–461.
14. N. Dyn and D. Levin, Bell-shaped basis functions for surface fitting, in *Approximation Theory and Applications* (Z. Zeigler, Ed.), pp. 113–129, Academic Press, New York, 1981.
15. W. E. L. Grimson, *From Images to Surfaces*, pp. 179–203, MIT Press, Cambridge, MA, 1981.
16. R. Franke, Scattered data interpolation: tests of some methods, *Math. Comput.* **38**, 1982, 181–200.
17. M. P. Do Carmo, *Differential Geometry of Curves and Surfaces*, pp. 197–209, Prentice–Hall, New York, 1976.
18. E. A. Lord and C. B. Wilson, *The Mathematical Description of Shape and Form*, pp. 133–135, Ellis-Horwood, Press, Chichester, England, 1984.
19. K. Ishii, Analytical shape determination for membrane structures, in *Proceedings World Congress on Space Enclosures, Building Res. Institute, Concordia University, Montreal*, 1976, pp. 137–149.
20. V. R. Southwell, *Relaxation Methods in Theoretical Physics*, Oxford Univ. Press, Oxford, 1946.