

# An Adaptive Clustering Algorithm for Image Segmentation

Thrasyvoulos N. Pappas

**Abstract**—The problem of segmenting images of objects with smooth surfaces is considered. The algorithm we present is a generalization of the  $K$ -means clustering algorithm to include spatial constraints and to account for local intensity variations in the image. Spatial constraints are included by the use of a Gibbs random field model. Local intensity variations are accounted for in an iterative procedure involving averaging over a sliding window whose size decreases as the algorithm progresses. Results with an eight-neighbor Gibbs random field model applied to pictures of industrial objects, buildings, aerial photographs, optical characters, and faces, show that the algorithm performs better than the  $K$ -means algorithm and its nonadaptive extensions that incorporate spatial constraints by the use of Gibbs random fields. A hierarchical implementation is also presented and results in better performance and faster speed of execution.

The segmented images are caricatures of the originals which preserve the most significant features, while removing unimportant details. They can be used in image recognition and as crude representations of the image. The caricatures are easy to display or print using a few grey levels and can be coded very efficiently. In particular, segmentation of faces results in binary sketches which preserve the main characteristics of the face, so that it is easily recognizable.

## I. INTRODUCTION

WE present a technique for segmenting a grey-scale image (typically 256 levels) into regions of uniform or slowly varying intensity. The segmented image consists of very few levels (typically 2–4), each denoting a different region, as shown in Fig. 1. It is a sketch, or caricature, of the original image which preserves its most significant features, while removing unimportant details. It can thus be the first stage of an image recognition system. However, assuming that the segmented image retains the intelligibility of the original, it can also be used as a crude representation of the image. The caricature has the advantage that it is easy to display or print with very few grey levels. The number of levels is crucial for special display media like paper, cloth, and binary computer screens. Also, the caricature can be coded very efficiently, since we only have to code the transitions between a few grey levels.

We develop an algorithm that separates the pixels in the image into clusters based on both their intensity and their

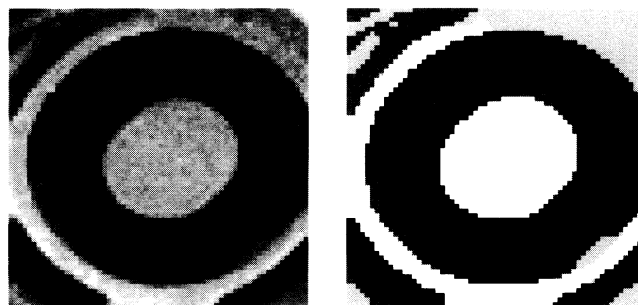


Fig. 1. Grey-scale image and its four-level segmentation.

relative location. The intensity of each region is assumed to be a slowly varying function plus noise. Thus we assume that we have images of objects with smooth surfaces, no texture. We make use of the spatial information by assuming that the distribution of regions is described by a Gibbs random field. The parameters of the Gibbs random field model the size and the shape of the regions.

A well-known clustering procedure is the  $K$ -means algorithm [1], [2]. When applied to image segmentation this approach has two problems: it uses no spatial constraints and assumes that each cluster is characterized by a constant intensity. A number of algorithms that were recently introduced by Geman and Geman [3], Besag [4], Derin and Elliot [5], Lakshmanan and Derin [6], and others, use Gibbs random fields and can be regarded as an extension of this algorithm to include spatial constraints. They all assume, however, that the intensity (or some other parameter) of each region is constant.

The technique we develop can be regarded as a generalization of the  $K$ -means clustering algorithm in two respects: it is adaptive and includes spatial constraints. Like the  $K$ -means clustering algorithm, our algorithm is iterative. Each region is characterized by a slowly varying intensity function. Given this intensity function, we define the *a posteriori* probability density function for the distribution of regions given the observed image. The density has two components, one constrains the region intensity to be close to the data and the other imposes spatial continuity. The algorithm alternates between maximizing the *a posteriori* probability density and estimating the intensity functions. Initially, the intensity functions are constant in each region and equal to the  $K$ -means cluster centers. As the algorithm progresses, the intensities are updated by averaging over a sliding window whose size

Manuscript received April 5, 1990; revised February 18, 1991.

The author is with the Signal Processing Research Department, AT&T Bell Laboratories, Murray Hill, NJ 07974.

IEEE Log Number 9106031.

progressively decreases. Thus the algorithm starts with global estimates and slowly adapts to the local characteristics of each region.

Our algorithm produces excellent caricatures of a variety of images. They include pictures of industrial objects, buildings, optical characters, faces, and aerial photographs. The binary sketches of faces, in particular, preserve the main characteristics of the face, so that it is easily recognizable. They can thus be used both as representations of the face and as input to a face recognition system.

The performance of our technique is clearly superior to  $K$ -means clustering for the class of images we are considering, as we show with several examples. It is also better than the nonadaptive extensions of  $K$ -means clustering that incorporate spatial constraints by the use of Gibbs random fields. Our experimental results indicate that the performance of the algorithm is also superior to existing region growing techniques (see [7] for a survey). In the two-level segmentation case our technique compares favorably to other adaptive thresholding techniques that can be found in [8], [9].

We also compare our adaptive clustering algorithm with the edge detector of [10], [11]. Its advantage over the edge detection approach is that it works with regions. In the edge detection case it is often difficult to associate regions with the detected boundaries. In our case regions are always well defined. The comparison becomes particularly interesting when we consider the case of faces. We will show that it is usually easier to recognize a face from a bilevel caricature than from its edges.

We conclude our presentation of the adaptive clustering algorithm by considering a hierarchical multigrid implementation. We construct a pyramid of different resolutions of an image by successively filtering and decimating by two, starting from the highest resolution image. The development relies on understanding the behavior of the algorithm at different levels of the resolution pyramid. The hierarchical implementation results in better performance and faster speed of execution.

The model for the class of images we are considering is presented in Section II. The algorithm is presented in Section III. In Section IV we examine the behavior of the algorithm on a variety of images. In Section V we consider a hierarchical implementation of the algorithm. The paper's conclusions are summarized in Section VI.

## II. MODEL

We model the grey-scale image as a collection of regions of uniform or slowly varying intensity. The only sharp transitions in grey level occur at the region boundaries. Let the observed grey scale image be  $y$ . The intensity of a pixel at location  $s$  is denoted by  $y_s$ , which typically takes values between 0 and 255. A segmentation of the image into regions will be denoted by  $x$ , where  $x_s = i$  means that the pixel at  $s$  belongs to region  $i$ . The number of different region types (or classes) is  $K$ . In this section we develop a model for the *a posteriori* probability den-

sity function  $p(x|y)$ . By Bayes' theorem

$$p(x|y) \propto p(y|x)p(x) \quad (1)$$

where  $p(x)$  is the *a priori* density of the region process and  $p(y|x)$  is the conditional density of the observed image given the distribution of regions.

We model the region process by a Markov random field. That is, if  $N_s$  is a neighborhood of the pixel at  $s$ , then

$$p(x_s|x_q, \text{ all } q \neq s) = p(x_s|x_q, q \in N_s). \quad (2)$$

We consider images defined on the Cartesian grid and a neighborhood consisting of the 8 nearest pixels. According to the Hammersley-Clifford theorem [12] the density of  $x$  is given by a Gibbs density which has the following form:

$$p(x) = \frac{1}{Z} \exp \left\{ - \sum_C V_C(x) \right\}. \quad (3)$$

Here  $Z$  is a normalizing constant and the summation is over all cliques  $C$ . A clique is a set of points that are neighbors of each other. The clique potentials  $V_C$  depend only on the pixels that belong to clique  $C$ .

Our model assumes that the only nonzero potentials are those that correspond to the one- and two-point cliques shown in Fig. 2. The two-point clique potentials are defined as follows:

$$V_C(x) = \begin{cases} -\beta, & \text{if } x_s = x_q \quad \text{and } s, q \in C \\ +\beta, & \text{if } x_s \neq x_q \quad \text{and } s, q \in C. \end{cases} \quad (4)$$

The parameter  $\beta$  is positive, so that two neighboring pixels are more likely to belong to the same class than to different classes. Increasing the value of  $\beta$  has the effect of increasing the size of the regions and smoothing their boundaries. The one-point clique potentials are defined as follows:

$$V_C(x) = \alpha^i, \quad \text{if } x_s = i \quad \text{and } s \in C, \quad \text{all } i. \quad (5)$$

The lower the value of  $\alpha^i$ , the more likely that pixel  $s$  belongs to class  $i$ . Thus, the parameters  $\alpha^i$  reflect our *a priori* knowledge of the relative likelihood of the different region types. In the remainder of this paper we will assume that all region types are equally likely and set  $\alpha^i = 0$  for all  $i$ . A more extensive discussion of Gibbs random fields can be found in [3]–[5].

The conditional density is modeled as a white Gaussian process, with mean  $\mu_s^i$  and variance  $\sigma^2$ .<sup>1</sup> Each region  $i$  is characterized by a different  $\mu_s^i$  which is a slowly varying function of  $s$ . Thus, the intensity of each region is modeled as a signal  $\mu_s^i$  plus white Gaussian noise with variance  $\sigma^2$ .

The combined probability density has the form

$$p(x|y) \propto \exp \left\{ - \sum_s \frac{1}{2\sigma^2} [y_s - \mu_s^{x_s}]^2 - \sum_C V_C(x) \right\}. \quad (6)$$

<sup>1</sup>Note that the superscript  $i$  in  $\mu_s^i$  and  $\alpha^i$  is an index, not an exponent. The few instances in the text where an exponent is used, as in  $\sigma^2$ , should be obvious from the context.

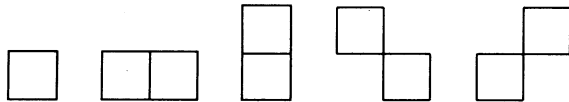


Fig. 2. Cliques for Gibbs probability density function.

We observe that the probability density function has two components. One constrains the region intensity to be close to the data. The other imposes spatial continuity.

Note that if the intensity functions  $\mu_s^i$  do not depend on  $s$ , then we get a model similar to those used in [3]–[5]. If we also set  $\beta = 0$ , then we get the  $K$ -means algorithm. Thus our technique can be regarded as a generalization of the  $K$ -means clustering algorithm. The Gibbs random field introduces the spatial constraints, and the pixel dependence of the intensity functions makes it adaptive. We found both of these additions to be essential for the proposed algorithm.

If we assume that the parameter  $\beta$  and the noise variance  $\sigma^2$  are known, then we must estimate both the distribution of regions  $x$  and the intensity functions  $\mu_s^i$ . This will be done in the following section. Note that the only constraint we impose on the functions  $\mu_s^i$  is that they are slowly varying. The estimation procedure we present in the following section uses this fact and implicitly imposes the constraint. The problem with using a parametric model (e.g., polynomial) for the region intensities is that, since the regions and intensity functions must be estimated concurrently, the parametric model becomes sensitive to inaccurate and fragmented region estimates.

Alternative models for the conditional density are possible. For example, Chalmond [13] and Dinten [14] use Besag's autonormal model [4] in the context of image restoration, and Bouman and Liu [15], [16] use an autoregressive model for segmentation of textured images.

We would like to point out that the Gibbs field is a good model of the region process only if we have a good model for the conditional density. The Gibbs model by itself is not very useful. It is easy to see that, in the absence of an observation, the optimal segmentation is just one region. In their discussion of the Ising model, Kindermann and Snell [17] observe that the Gibbs field is ill behaved if there is no strong outside force (observed image with edges, in our case).

### III. ALGORITHM

In this section we consider an algorithm for estimating the distribution of regions  $x$  and the intensity functions  $\mu_s^i$ . Note that the functions  $\mu_s^i$  are defined on the same grid as the original grey-scale image  $y$  and the region distribution  $x$ . Like the  $K$ -means clustering algorithm, our algorithm is iterative. It alternates between estimating  $x$  and the intensity functions  $\mu_s^i$ .

First, we consider the problem of estimating the local intensity functions. Given the region labels  $x$ , we estimate the intensity  $\mu_s^i$  at each pixel  $s$  as follows. We average the

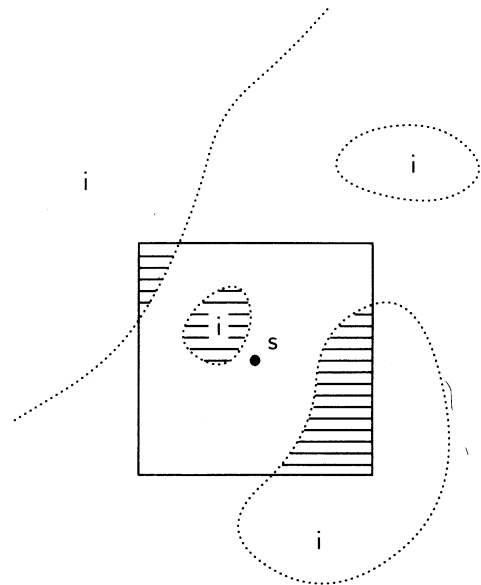


Fig. 3. Local average estimation.

grey levels of all the pixels that belong to region  $i$  and are inside a window of width  $W$  centered at pixel  $s$ , as shown in Fig. 3. When the number of pixels of type  $i$  inside the window centered at  $s$  is too small, the estimate  $\mu_s^i$  is not very reliable. At these points the intensity  $\mu_s^i$  is undefined. When this happens,  $x_s$  cannot be assigned to level  $i$ . Thus we have to pick the minimum number of pixels  $T_{\min}$  that are necessary for estimating  $\mu_s^i$ . The higher this threshold the more robust the computation of  $\mu_s^i$ . On the other hand, any small isolated regions with area less than this threshold may disappear. Thus the presence of this threshold strengthens the spatial constraints. A reasonable choice for the minimum number of pixels necessary for estimating  $\mu_s^i$  is  $T_{\min} = W$ , the window width. This value of the threshold guarantees that long one-pixel-wide regions will be preserved.

We have to obtain estimates of  $\mu_s^i$  for all region types  $i$  and all pixels  $s$ . Clearly, the required computation is enormous, especially for large window sizes. Instead, we compute the estimates  $\mu_s^i$  only on a grid of points and use bilinear interpolation to obtain the remaining values. The spacing of the grid points depends on the window size. We chose the spacing equal to half the window size in each dimension (50% overlap). Fig. 4 shows a window and the location of the centers of adjacent windows. The functions  $\mu_s^i$  are smooth and therefore the interpolated values are good approximations of the values we would obtain by an exact calculation. Note that the larger the window the smoother the functions, hence the spacing of the grid points increases with the window size. Note also that, because of the dependence of the grid spacing on the window size, the amount of computation is independent of window size.

Fig. 5 shows an original image, its two-level segmentation, and the local intensity functions for the two levels.

Second, we consider the problem of estimating the distribution of regions. Given the intensity functions  $\mu_s^i$ , we

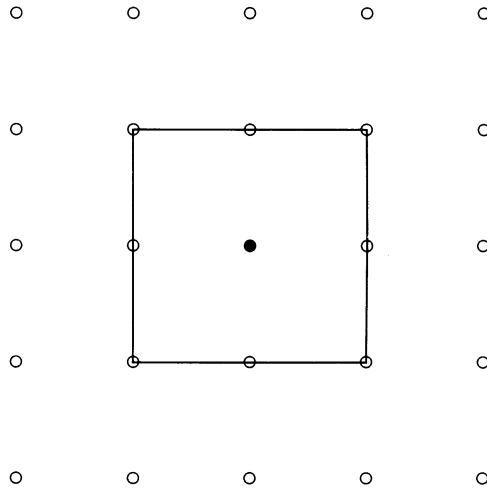


Fig. 4. Window locations for local average estimation.

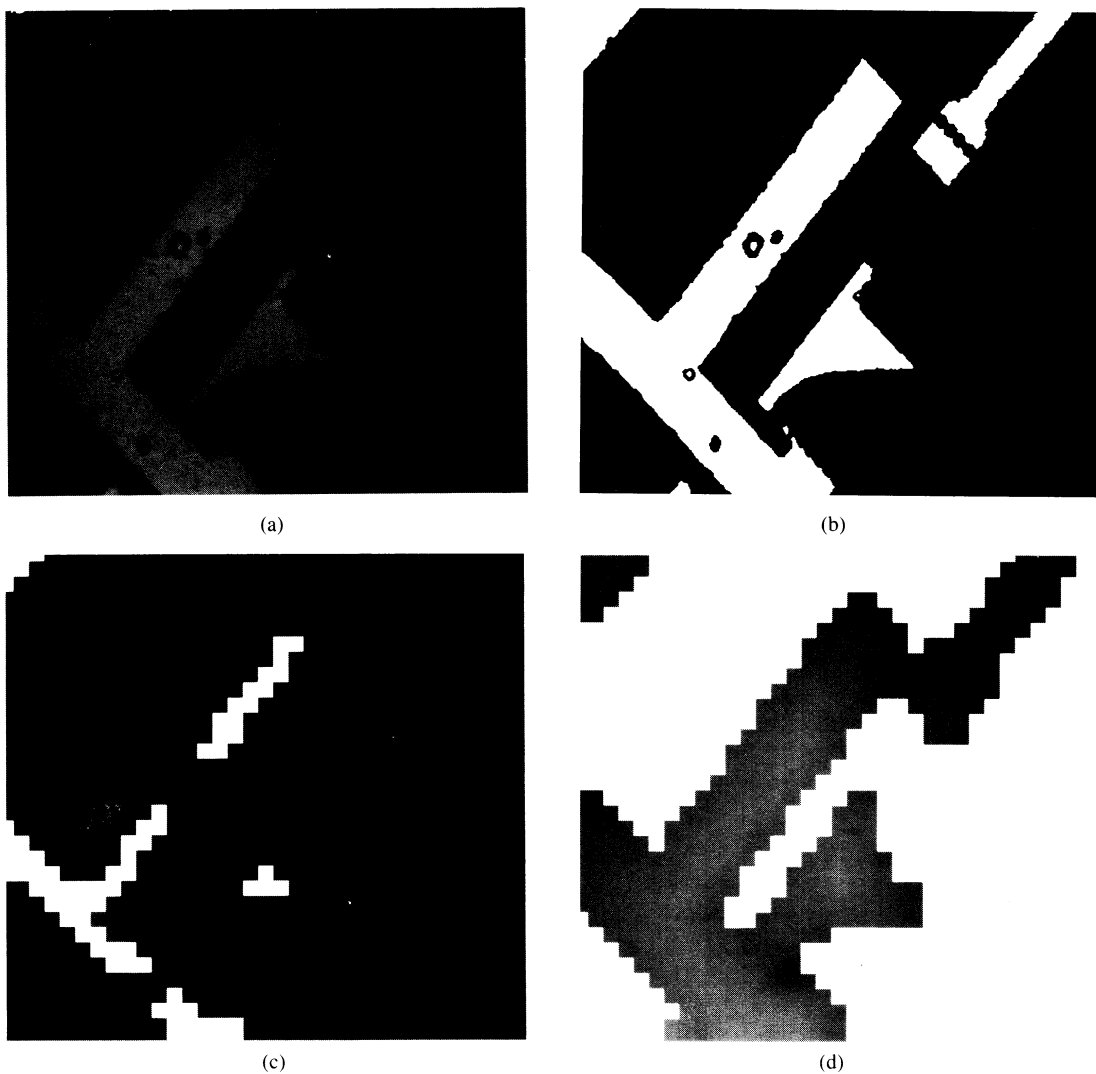


Fig. 5. Local intensity functions. (a) Original image. (b) Two-level segmentation. (c) Level 0 intensity function. (d) Level 1 intensity function.

want to maximize the *a posteriori* probability density (6) to obtain the MAP estimate of  $x$ . Finding the global maximum of this function requires a lot of computations. It

can be done using the Gibbs sampler and simulated annealing [3]. Instead, we maximize the conditional density at each point  $x_s$  given the data  $y$  and the current  $x$  at all

other points:

$$\begin{aligned} p(x_s | y, x_q, \text{all } q \neq s) \\ &= p(x_s | y_s, x_q, q \in N_s) \\ &\propto \exp \left\{ -\frac{1}{2\sigma^2} [y_s - \mu_s^{x_s}]^2 - \sum_{x_s \in C} V_C(x) \right\}. \quad (7) \end{aligned}$$

The equality on the left follows from the Markovian property and the whiteness of the noise. The maximization is done at every point in the image and the *cycle* is repeated until convergence. This is the iterated conditional modes (ICM) approach proposed by Besag [4], sometimes also called the greedy algorithm [18]. It corresponds to instantaneous freezing in simulated annealing. Bouman and Liu use this greedy algorithm in their multiple resolution approach for image segmentation [15], [16]. The order in which the points are updated within a cycle depends on the computing environment [4]. In our implementation the updating was done in a raster scan. This procedure converges to a local maximum [4]. It stops when the number of pixels that change during a cycle is less than a threshold (typically  $M/10$  for an  $M \times M$  image). The convergence is very rapid, usually in less than five cycles, consistent with Besag's observations.

Now we consider the overall adaptive clustering algorithm. We obtain an initial estimate of  $x$  by the  $K$ -means algorithm [2]. Starting from this segmentation, our algorithm alternates between estimating  $x$  and estimating the intensity functions  $\mu_s^i$ . We define an *iteration* to consist of one update of  $x$  and one update of the functions  $\mu_s^i$ . The window size for the intensity function estimation is kept constant until this procedure converges, usually in less than ten iterations (The number of iterations could vary with image content.) Our stopping criterion is that the last iteration converges in one cycle. The whole procedure is then repeated with a new window size. The adaptation is achieved by varying the window size  $W$ . Initially, the window for estimating the intensity functions is the whole image and thus the intensity functions of each region are constant. As the algorithm progresses, the window size decreases. The reason for this is that in the early stages of the algorithm, the segmentation is crude, and a large window is necessary for robust estimation of the intensity functions. As the algorithm progresses, the segmentation becomes better, and smaller windows give more reliable and accurate estimates. Thus the algorithm, starting from global estimates, slowly adapts to the local characteristics of each region. A detailed flowchart of our algorithm is given in Fig. 6. The algorithm stops when the minimum window size is reached. Typically we keep reducing the window size by a factor of two, until a minimum window size of  $W = 7$  pixels.

If the window size is kept constant and equal to the whole image, then the result is the same as in the approach of [3]–[6]. This nonadaptive clustering algorithm is only an intermediate result of our adaptive clustering algorithm. It results in a segmentation similar to the  $K$ -

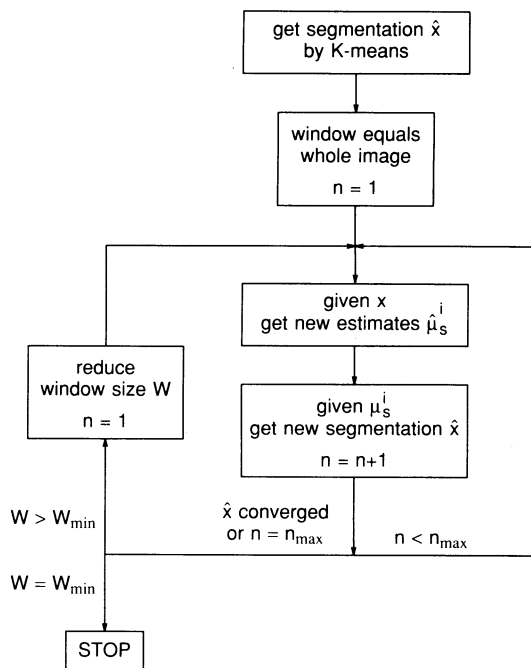


Fig. 6. Adaptive clustering algorithm.

means; only the edges are smoother. The strength of the proposed approach lies in the variation of the window size which allows the intensity functions to adjust to the local characteristics of the image. Note that the estimation procedure imposes the smoothness constraint on the local intensity functions. Also, there is no need to split our regions into a set of connected pixels as is done in [19]. The variable window size takes care of this problem.

We assume that the noise variance  $\sigma^2$  is known or can be estimated independently of the algorithm. This is a reasonable assumption in many cases, especially when we have images obtained under similar conditions. From (6) it follows that increasing  $\sigma^2$  is equivalent to increasing  $\beta$ . In fact, the parameter  $\beta$  of the Gibbs random field can be chosen so that the region model is weaker in the early stages (algorithm follows the data), and stronger in the later stages (algorithm follows model) [4], [20]. In our implementations we chose the constant value  $\beta = 0.5$  for all the iterations. Once the algorithm converges at the smallest window size, then  $\beta$  could be increased to obtain smoother segment contours. Thus, given an image to be segmented, the most important parameter to choose is the noise variance. In Section V we show that the performance of the algorithm is reasonable over a wide range of noise variances. In fact, the noise variance controls the amount of detail detected by the algorithm.

We also have to choose the number of different regions  $K$ . We found that  $K = 4$  works well for most images. Note that, in contrast to the  $K$ -means algorithm and some of the other adaptive schemes [7], [8], the choice of  $K$  is not critical because the adaptation of the intensity functions allows the same region type to have different intensities in different parts of the image. In the following section we discuss the choice of  $K$  more extensively.

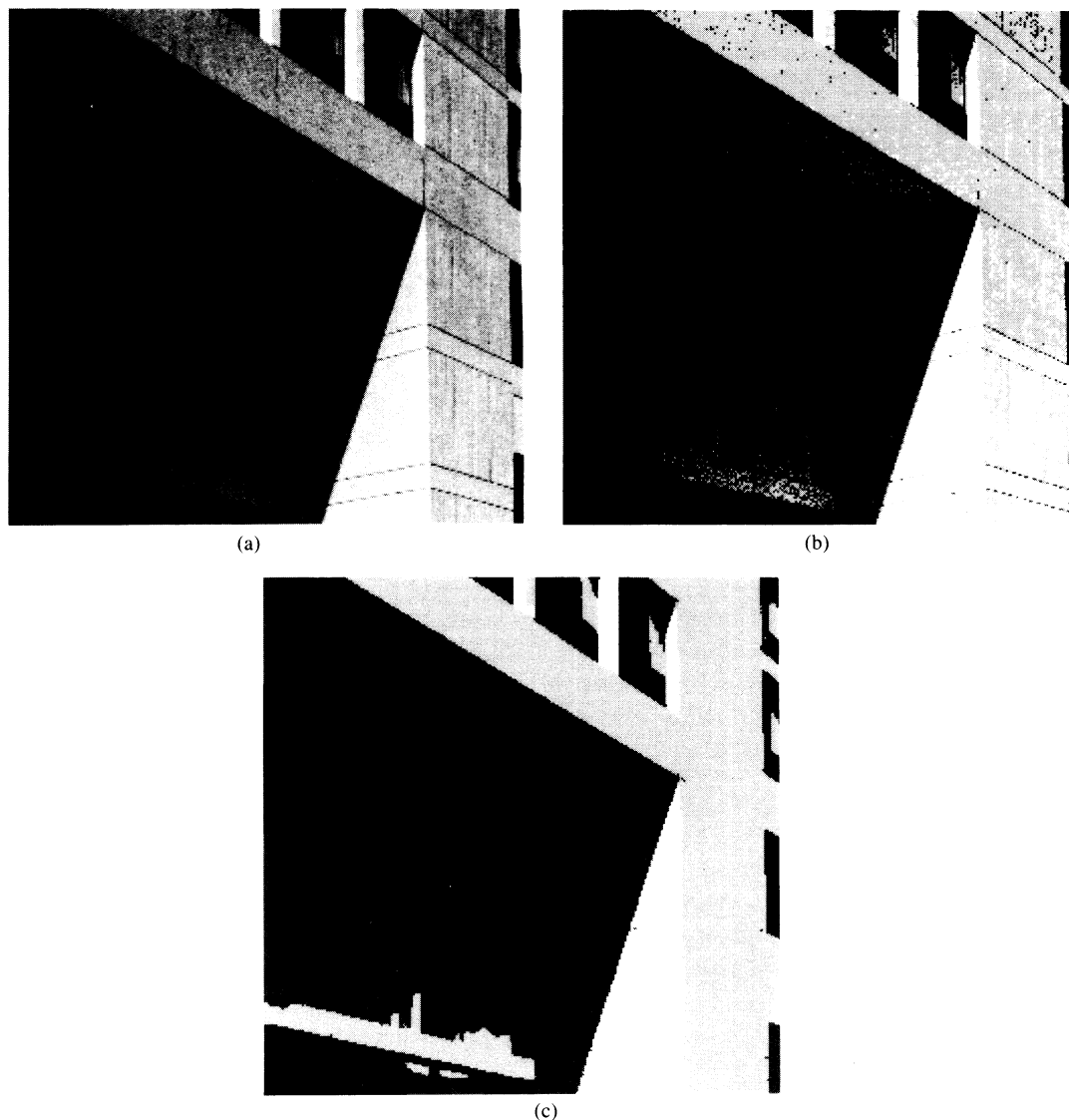


Fig. 7. Example A. (a) Original image. (b)  $K$ -means algorithm ( $K = 4$  levels). (c) Adaptive clustering ( $K = 4$  levels).

The amount of computation for the estimation of  $x$  depends on the image dimensions, the number of different windows used, the number of classes and, of course, image content. As we pointed out earlier, the amount of computation for each local intensity calculation does not depend on the window size. We should also point out that both the local intensity calculation and each cycle of the probability maximization can be done in parallel. Thus, even though the algorithm is very slow on serial machines, it can be considerably faster on a parallel computer.

#### IV. EXAMPLES

In this section we examine the performance of the algorithm on a variety of images. We discuss the choice of the number of regions  $K$ , and compare adaptive clustering to  $K$ -means clustering and edge detection.

An original image is shown in Fig. 7(a); the resolution is  $256 \times 256$  pixels and the grey-scale range is 0 to 255

(8 b). White Gaussian noise with  $\sigma = 7$  grey levels has been added to this image. Fig. 7(b) shows the result of the  $K$ -means algorithm for  $K = 4$  levels. The two problems we mentioned earlier are apparent. There are a lot of isolated pixels and areas of uniform intensity (e.g., walls on lower right) are not well separated. Fig. 7(c) shows the result of our adaptive algorithm for  $K = 4$  levels. The superior performance of the adaptive algorithm is apparent. Both problems have been eliminated. We can understand the limitations of the  $K$ -means algorithm when we look at a scan line of the image. The dotted line in Fig. 8(a) shows a horizontal scan line in the bottom half of the image of Fig. 7(a), and the solid lines show the four characteristic levels selected by the  $K$ -means algorithm. The solid line of Fig. 8(b) shows the resulting segmentation. We observe that while  $\mu^0$  and  $\mu^1$  are reasonable models of the local image intensity,  $\mu^2$  and  $\mu^3$  are not. This is because the selection of the four levels is based on the histogram of the whole image and does not

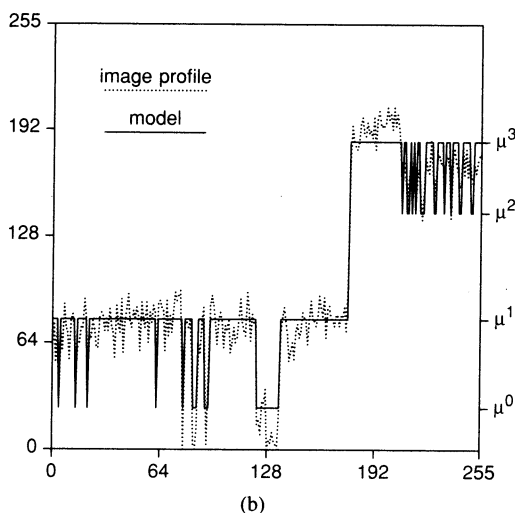
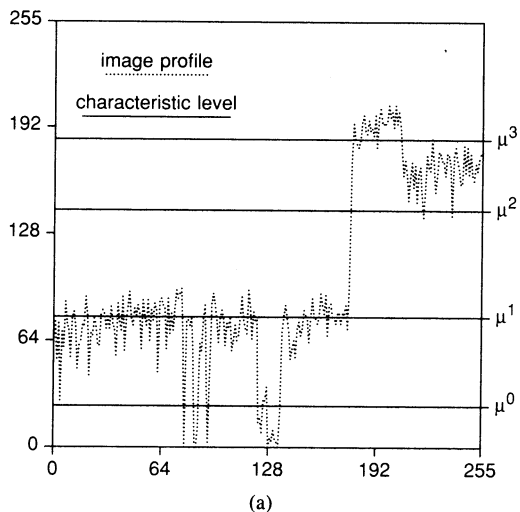


Fig. 8.  $K$ -means algorithm. (a) Image profile and characteristic levels. (b) Image profile and model.

necessarily match the local characteristics. Fig. 9(a) shows the intensity functions of the adaptive algorithm and Fig. 9(b) shows the corresponding segmentation. The intensity functions have indeed adapted to the local intensities of the image.

Another original image is shown in Fig. 10(a); the resolution is  $244 \times 256$  pixels and the grey-scale range is 0 to 255. The standard deviation of the noise was estimated to be  $\sigma = 7$  grey levels. Fig. 10(b) shows the result of the  $K$ -means algorithm. Fig. 10(c) shows the result of our adaptive algorithm. Again we chose  $K = 4$ . The problems we mentioned above are again evident in the  $K$ -means result. The adaptive algorithm gives a much better segmentation, even though some problems are still present (e.g., the rings of the wrench in the lower right are not well defined). Overall, we could say that the adaptive algorithm produces a simple representation of the image, while retaining most of the important information.

One of the important parameters in the adaptive clustering algorithm is the number of classes  $K$ . In most clustering algorithms the choice of the number of classes is important and could have a significant effect on the qual-

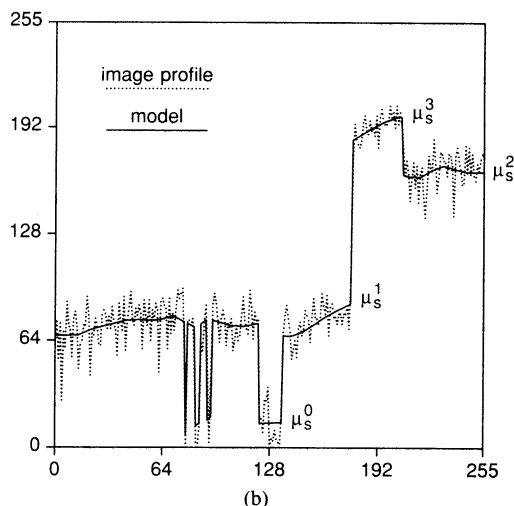
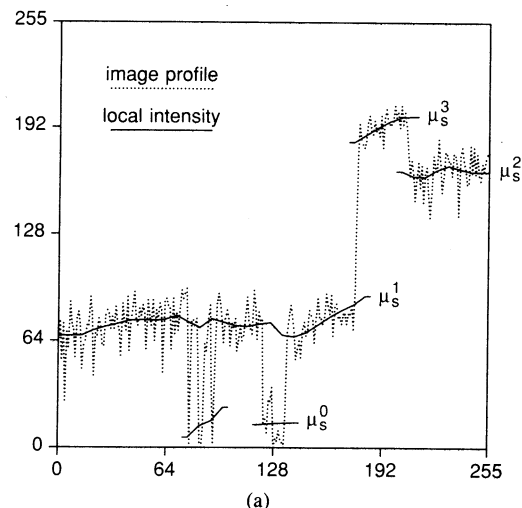


Fig. 9. Adaptive clustering algorithm. (a) Image profile and local intensity functions ( $15 \times 15$  window). (b) Image profile and model ( $15 \times 15$  window).

ity of segmentation. Thus, many authors consider estimating of the number of classes as a part of their segmentation algorithms, for example [21], [16]. In the case of the  $K$ -means algorithm, choosing the wrong number of classes can be disastrous as we will see in the following examples. However, our adaptive algorithm is quite robust to the choice of  $K$ . This is because the characteristic levels of each class adapt to the local characteristics of the image, and thus regions of entirely different intensities can belong to the same class, as long as they are separated in space.

Fig. 11(a) shows the result of the  $K$ -means algorithm for  $K = 2$ . Because of the wide variation of the grey-scale intensity throughout the image, a lot of detail has been lost in the  $K$ -means case. Fig. 11(b) shows the result of our adaptive algorithm for  $K = 2$ . Here most of the detail in the image has been recovered.<sup>2</sup> We can argue that the

<sup>2</sup>Note that the thin oblique lines of Figs. 11(b) and 7(b) have disappeared in Figs. 11(b) and 7(b), respectively. This is caused by the combination of the Gibbs constraints and the threshold  $T_{\min}$  defined in Section III. The only one-pixel-wide features preserved are those of high contrast.

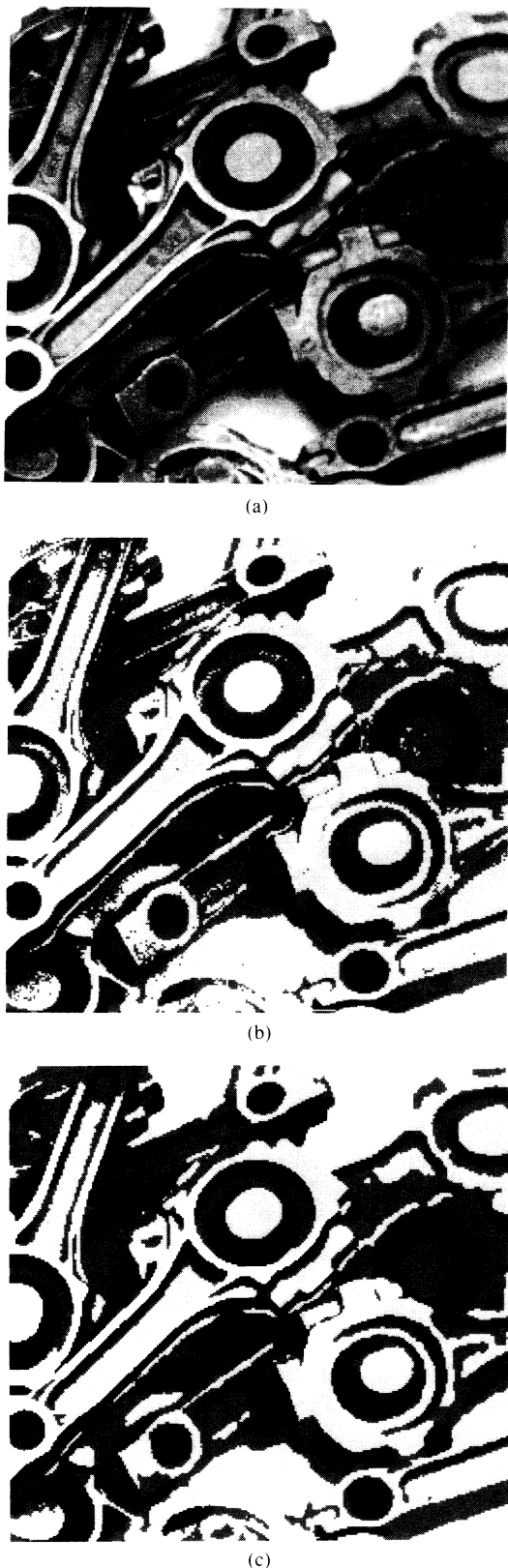


Fig. 10. Example B. (a) Original image. (b)  $K$ -means algorithm ( $K = 4$  levels). (c) Adaptive clustering ( $K = 4$  levels).

segmentation is not perfect; indeed, some edges are missing (e.g., the sky in the upper right) and some spurious edges are present (e.g., around the windows in the lower left). Clearly,  $K = 4$  results in a better segmentation.

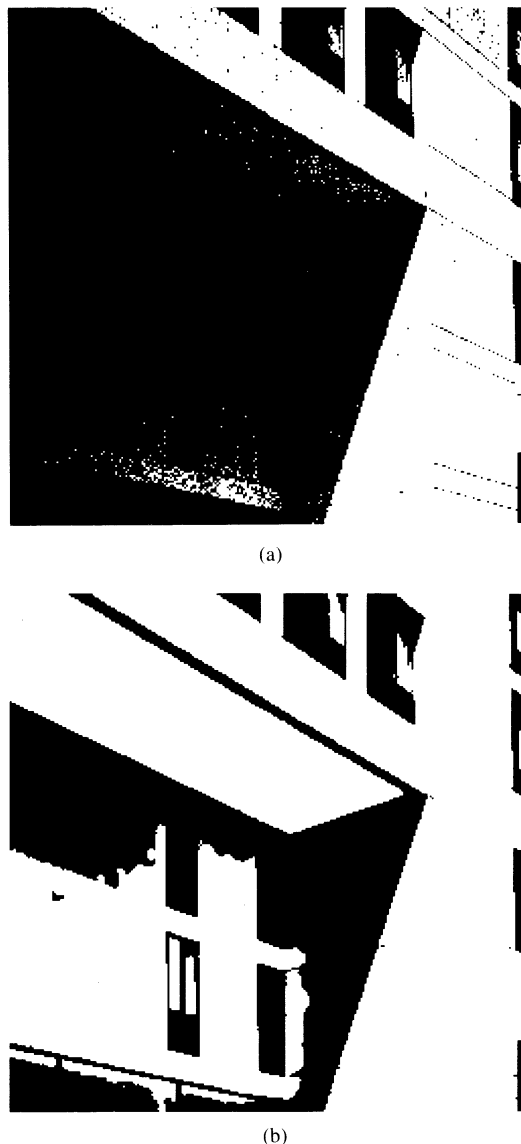


Fig. 11. Example A. (a)  $K$ -means algorithm ( $K = 2$  levels). (b) Adaptive clustering ( $K = 2$  levels).

In many images  $K = 4$  is the best choice. This appears to be related to the four-color theorem [22], even though a proof would be very difficult. However, for  $K = 2$  the adaptive algorithm results in a very good representation of the image, even though the individual segments don't always correspond to actual edges in the image. The value of  $K = 2$  is especially important because it simplifies both the display (e.g., on paper or on binary computer screens) and coding of the image.

As another example, consider the original image shown in Fig. 12(a); the resolution is  $512 \times 512$  pixels and the grey-scale range is 0 to 255. The standard deviation of the noise was assumed to be  $\sigma = 4$  grey levels. Fig. 12(b) shows the result of the  $K$ -means algorithm for  $K = 2$ . Fig. 12(c) shows the result of our adaptive algorithm. The advantage of the adaptive algorithm is apparent. For example, an airstrip at the lower left of the image has disappeared in the  $K$ -means case, but is nicely preserved by the adaptive algorithm.



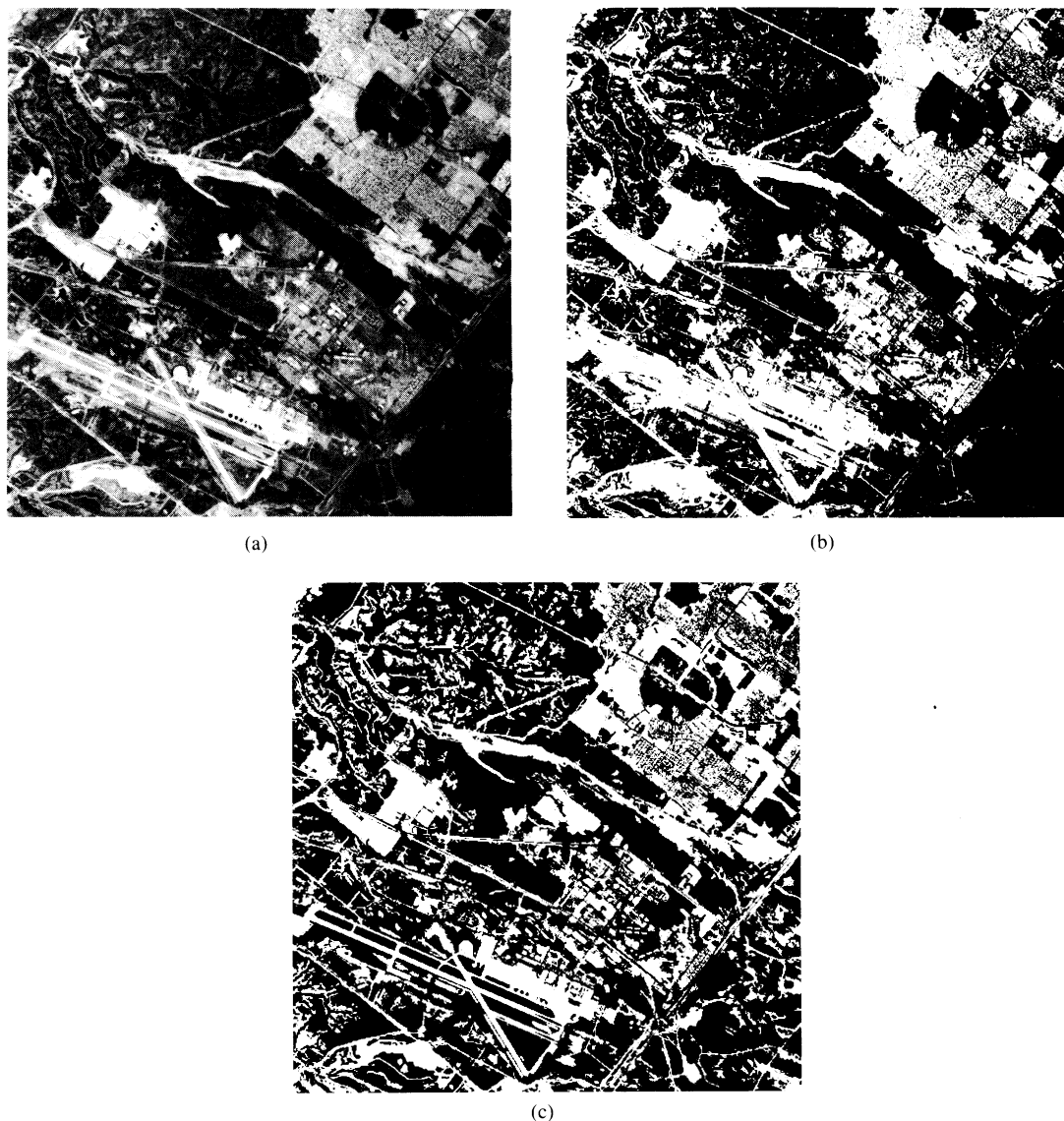


Fig. 12. Example C. (a) Original image. (b)  $K$ -means algorithm ( $K = 2$  levels). (c) Adaptive clustering ( $K = 2$  levels).

In some cases  $K$  is constrained by the statement of the problem. For example, consider the image of Fig. 13(a). Here  $K = 2$  since each point of the image either belongs to the character or to the background. The resolution is  $64 \times 64$  pixels and the grey-scale range is 0 to 255. This image was obtained by blurring the optical character “Æ” with a Gaussian filter (the standard deviation is 2.4 pixels) and adding white Gaussian noise with  $\sigma = 10$  grey levels. Fig. 13(b) shows the result of the  $K$ -means algorithm. Fig. 13(c) shows the result of our adaptive algorithm. We observe that the adaptive algorithm recovers the hole in the center of the character. This could be crucial in the correct interpretation of the character. Thus our algorithm could be an important component of an optical character recognition system.

The performance of the two-level adaptive algorithm is particularly good on grey-scale images of human faces. Consider the original image shown in Fig. 14(a); the resolution is  $512 \times 512$  pixels and the grey-scale range is 0

to 255. The standard deviation of the noise was assumed to be  $\sigma = 7$  grey levels. Fig. 14(b) shows the result of the  $K$ -means algorithm and Fig. 14(c) shows the result of our adaptive algorithm. The  $K$ -means algorithm produces an image that is too light and, therefore, barely gets some of the facial characteristics. The adaptive algorithm, on the other hand, gives a very good sketch of the face. Note that the nose is implied by a couple of black regions at its base, much like an artist might sketch it with a few brush strokes. In general, the  $K$ -means algorithm produces a binary image that may be too light or too dark, depending on the overall image content. The proposed algorithm adapts to the local characteristics and preserves the features of the face.

The binary sketches preserve the characteristics of the human faces as we have observed over several examples obtained under different conditions (see Section V for more examples). As we discussed, the sketches can be used for displaying or printing on special media like pa-

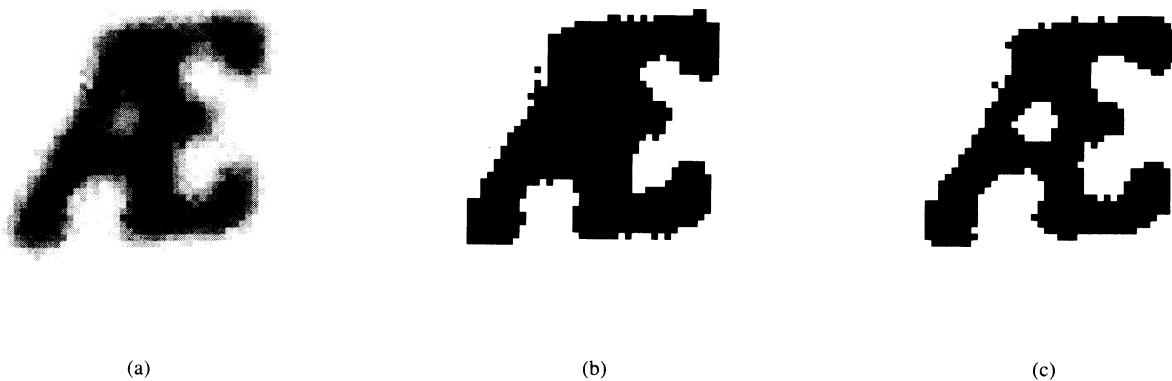


Fig. 13. Example D. (a) Original image. (b)  $K$ -means algorithm ( $K = 2$  levels). (c) Adaptive clustering ( $K = 2$  levels).

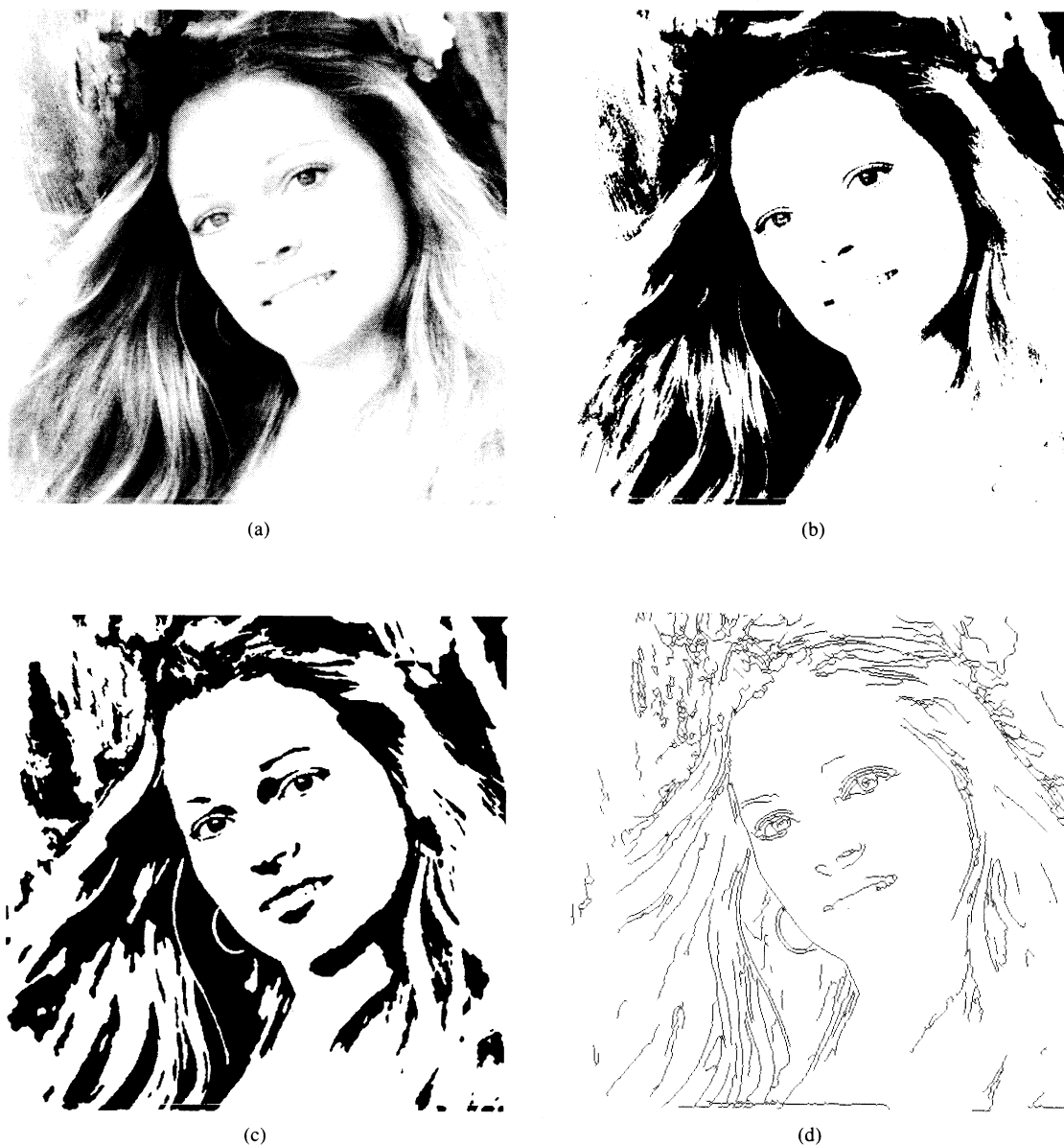


Fig. 14. Example E. (a) Original image. (b)  $K$ -means algorithm ( $K = 2$  levels). (c) Adaptive clustering ( $K = 2$  levels). (d) Edge detection.

per, cloth and binary computer screens. They can be coded with very few bits of data, since we only have to code the transitions between a few grey levels. In [23] we show that coding of typical face caricatures requires less than 0.1 b/pixel, which is substantially lower than what waveform coding techniques require.

The dual to image segmentation is edge detection. It is thus interesting to compare our segmentation results with an edge detector. Here we use the edge detector of Boie and Cox [10], [11]. Fig. 14(d) shows the result of the edge detector with integrated line detection; their algorithm includes an automatic threshold selection. The algorithm is very successful in detecting the most important edges. We observe, however, that it is not as easy to recognize the person from the edges as it is from the binary sketches. This demonstrates the advantage of representing an image as a set of black and white regions, as opposed to a set of edges. Note the basic tradeoff between region segmentation and edge detection. As we mentioned, not all region transitions correspond to edges in the image. On the other hand, the detected edges do not necessarily define closed regions.

As an indication of the amount of computation required by the adaptive clustering algorithm, we look at the running time on a SUN SPARC station 1+. The computation time for the four-level segmentation of Fig. 7(c) was 11 min of user time. The  $K$ -means clustering of Fig. 7(b) required only 10 s of user time. The computation time for the two-level segmentation of Fig. 11(b) was 7.6 min of user time and for the  $K$ -means clustering of Fig. 11(a) 4 sec of user time. The resolution of these images is  $256 \times 256$  pixels. When the resolution increases to  $512 \times 512$  pixels, as in Fig. 12, the computation times become 35 min for the adaptive algorithm and 13 s for the  $K$ -means algorithm. In our implementations we made no systematic attempt to make the code more efficient, and the choice of termination criteria was conservative.

## V. HIERARCHICAL MULTIGRID IMPLEMENTATION

In this section we present a hierarchical implementation of our segmentation algorithm. The goal is to improve both its performance and speed of execution. The development of such a hierarchical approach relies on understanding the behavior of the algorithm at different levels of the resolution pyramid. In particular, we would like to know how the various parameters of the algorithm change with the resolution of the image.

We construct a pyramid of images at different resolutions in the following way. Starting from the highest resolution image, we obtain the next image in the pyramid by filtering and decimating by a factor of two. The low-pass decimation filters we use have a sharp cutoff so that the sharpness of the edges is preserved at the lower resolutions.

The image model we described in Section II can be written as follows:

$$y_s = I_s + n_s \quad (8)$$

where  $I_s$  is the piecewise continuous signal and  $n_s$  is white Gaussian noise with standard deviation  $\sigma$ . The signal at the next resolution has the same form

$$y'_s = I'_s + n'_s \quad (9)$$

where  $I'_s$ , the filtered and decimated  $I_s$ , is approximately piecewise continuous and  $n'_s$ , the filtered and decimated  $n_s$ , is approximately white Gaussian with standard deviation  $\sigma' = \sigma/2$ . Thus, the model is still valid at the next resolution and only the noise standard deviation  $\sigma$  changes by a factor of two.

Based on our previous discussion of the role of the parameter  $\beta$ , it is reasonable to assume that it does not change with resolution. Bouman and Liu [15], [16] adopt a similar approach in their multiple resolution segmentation of textured images. An interesting alternative approach for the choice of the Gibbs parameters is proposed by Gidas [24], who obtains an approximation of  $\beta$  at different resolutions based on a probabilistic model for the whole pyramid. Given our choice of  $\beta$  and  $\sigma$ , the only remaining parameter is the detection threshold  $T_{\min}$ . We assume that  $T_{\min} = W$ , the window width, is a reasonable choice for the detection threshold  $T_{\min}$  at the highest level. As we lower the resolution we expect that significant features become smaller and, therefore, we should reduce the detection threshold  $T_{\min}$ . We decrease  $T_{\min}$  by a factor of two from one resolution to the next.

The hierarchical implementation of the algorithm is described in Fig. 15. We start at the lowest resolution image, and apply our segmentation algorithm as described in the previous sections, starting from the  $K$ -means clusters. When the minimum window size is reached, we move to the next level in the pyramid and use the current segmentation, expanded by a factor of two, as a starting point. Since we now have a good starting point, we need not iterate through the whole sequence of windows. We start where we left off at the previous resolution, which is twice the minimum window size of that resolution.

This approach reduces the amount of computation significantly, since most iterations are at the lowest level of resolution. The improvement in performance is also significant as the following example indicates. In our implementations we used a four-level pyramid.

An original image is shown in Fig. 16(a); the resolution is  $512 \times 512$  pixels. We assume that the noise level of this image is  $\sigma = 8$  grey levels. Fig. 16(b) shows the result of the adaptive algorithm for  $K = 2$  levels, applied to the full-scale image. Fig. 16(c) shows the result of the hierarchical implementation of the two-level adaptive algorithm. Both result in very good segmentations. We can argue, however, that the hierarchical segmentation has better adjusted to the local image characteristics, as can be seen by the difference in the lips and the area above the hat.

As an indication of the potential computational savings of the hierarchical implementation we look at the running times of the two algorithms on a SUN SPARC station 1+. The full-scale implementation of Fig. 16(b) required 36

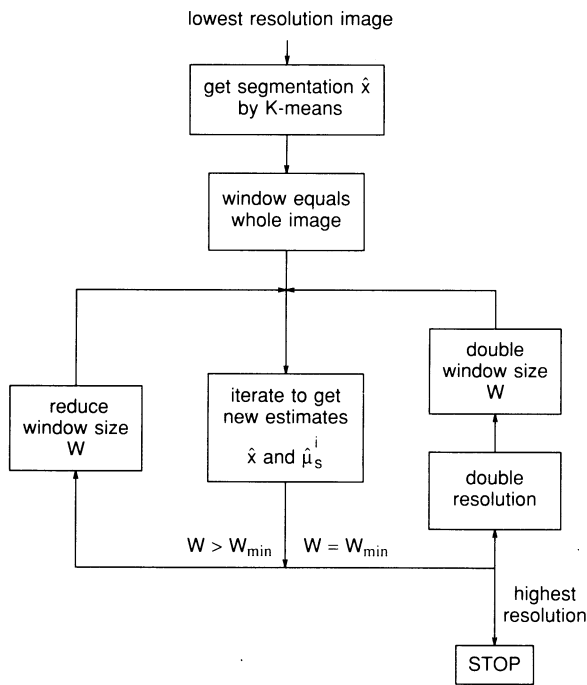


Fig. 15. Hierarchical adaptive clustering algorithm.

min of user time, while the hierarchical implementation of Fig. 16(c) required only 8.2 min of user time. The first (lowest resolution) stage required 10 s of user time, the second 16 s, the third 118 s, and the last 348 s. Again, no systematic attempt was made in our implementations to improve the efficiency of the code.

We now look at the performance of the algorithm for different noise-level estimates. An original image is shown in Fig. 17(a); the resolution is  $512 \times 512$  pixels. Figs. 17(b)–(e), show the result of the hierarchical implementation of the adaptive algorithm ( $K = 2$ ) for noise level estimates  $\sigma = 4, 8, 16,$  and  $32$  grey levels, respectively. The original image is the same in all cases. Observe how the amount of detail detected by the algorithm decreases as the noise variance increases. This example illustrates that the algorithm's performance is reasonable over a wide range of noise variances and, moreover, that we can control the amount of detail by choosing the noise variance. Thus, our knowledge of the level of noise determines the amount of detail in the segmented image.

## VI. CONCLUSION

We presented an algorithm for segmenting images of objects with smooth surfaces. We used Gibbs random fields to model the region process. The parameters of the Gibbs random field model the size and the shape of the regions. The intensity of each region is modeled as a slowly varying function plus white Gaussian noise. The adaptive technique we developed can be regarded as a generalization of the  $K$ -means clustering algorithm. We applied our algorithm to pictures of industrial objects, buildings, aerial photographs, faces, and optical characters. The segmented images are useful for image display,



(a)



(b)



(c)

Fig. 16. Example F. (a) Original image. (b) Adaptive clustering ( $K = 2$ ). (c) Hierarchical adaptive clustering ( $K = 2$ ).



(a)



(b)



(c)



(d)



(e)

Fig. 17. Example G: Hierarchical adaptive clustering ( $K = 2$ ). (a) Original image. (b)  $\sigma = 4$ . (c)  $\sigma = 8$ . (d)  $\sigma = 16$ . (e)  $\sigma = 32$ .

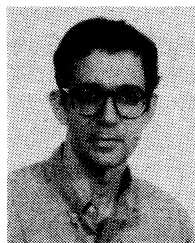
coding, and recognition. We believe that our approach can be extended to the case of images that include regions with slowly varying textures, as well as moving images.

#### ACKNOWLEDGMENT

The author would like to thank I. Cox for providing the code for the Boie-Cox edge detector.

#### REFERENCES

- [1] J. T. Tou and R. C. Gonzalez, *Pattern Recognition Principles*. Reading MA: Addison-Wesley, 1974.
- [2] R. M. Gray and Y. Linde, "Vector quantizers and predictive quantizers for Gauss-Markov sources," *IEEE Trans. Commun.*, vol. COM-30, no. 2, pp. 381-389, Feb. 1982.
- [3] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI-6, no. 6, pp. 721-741, Nov. 1984.
- [4] J. Besag, "On the statistical analysis of dirty pictures," *J. Roy. Stat. Soc. B*, vol. 48, no. 3, pp. 259-302, 1986.
- [5] H. Derin and H. Elliot, "Modeling and segmentation of noisy and textured images using Gibbs random fields," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI-9, no. 1, pp. 39-55, Jan. 1987.
- [6] S. Lakshmanan and H. Derin, "Simultaneous parameter estimation and segmentation of Gibbs random fields using simulated annealing," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 11, no. 8, pp. 799-813, Aug. 1989.
- [7] R. M. Haralick and L. G. Shapiro, "Image segmentation techniques," *Comput. Vision, Graph., Image Processing*, vol. 29, pp. 100-132, 1985.
- [8] P. K. Sahoo, S. Soltani, and A. K. C. Wong, "A survey of thresholding techniques," *Comput. Vision, Graph., Image Processing*, vol. 41, pp. 233-260, 1988.
- [9] J. S. Weszka, "A survey of threshold selection techniques," *Comput. Graph. Image Processing*, vol. 7, pp. 259-265, 1978.
- [10] R. A. Boie and I. J. Cox, "Two-dimensional optimum edge recognition using matched and Wiener filters for machine vision," in *Proc. IEEE First Int. Conf. Comput. Vision* (London, England), June 8-11, 1987, pp. 450-456.
- [11] I. J. Cox, R. A. Boie, and D. A. Wallach, "Line recognition," in *Proc. Tenth Int. Conf. Patt. Recogn.* (Atlantic City, NJ), June 16-21, 1990, pp. 639-645.
- [12] J. Besag, "Spatial interaction and the statistical analysis of lattice systems," *J. Roy. Stat. Soc. B*, vol. 26, no. 2, pp. 192-236, 1974.
- [13] B. Chalmond, "Image restoration using an estimated Markov model," *Signal Processing*, vol. 15, no. 2, pp. 115-129, Sept. 1988.
- [14] J. M. Dinten, "Tomographic reconstruction of axially symmetric objects: Regularization by a Markovian modelization," presented at the 10th Int. Congr. Patt. Recog., 1990.
- [15] C. Bouman and B. Liu, "Segmentation of textured images using a multiple resolution approach," in *Proc. ICASSP-88* (New York), Apr. 1988, pp. 1124-1127.
- [16] C. Bouman and B. Liu, "Multiple resolution segmentation of textured images," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 13, no. 2, pp. 99-113, Feb. 1991.
- [17] R. Kindermann and J. L. Snell, *Markov Random Fields and their Applications*. American Mathematical Society, 1980.
- [18] H. Derin, "Experimentations with simulated annealing in allocation optimization problems," in *Proc. CISS-86*, pp. 165-171.
- [19] T. Aach, U. Franke, and R. Mester, "Top-down image segmentation using object detection and contour relaxation," in *Proc. ICASSP-89* (Glasgow, U.K.), May 1989, pp. 1703-1706.
- [20] G. Wolberg and T. Pavlidis, "Restoration of binary images using stochastic relaxation with annealing," *Patt. Recogn. Lett.*, vol. 3, no. 6, pp. 375-387, Dec. 1985.
- [21] J. Zhang and J. W. Modestino, "A model-fitting approach to cluster validation with application to stochastic model-based image segmentation," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 12, no. 10, pp. 1009-1017, Oct. 1990.
- [22] K. Appel and W. Haken, "Every planar map is four colorable: Parts I and II," *Ill. J. Math.*, vol. 21, no. 3, pp. 429-567, Sept. 1977.
- [23] T. N. Pappas, "Adaptive thresholding and sketch-coding of grey level images," *Proc. SPIE Int. Soc. Opt. Eng.*, vol. 1199, Nov. 1989.
- [24] B. Gidas, "A renormalization group approach to image processing problems," *IEEE Trans. Patt. Anal. Machine Int.*, vol. 11, no. 2, pp. 164-180, Feb. 1989.



**Thrasyvoulos N. Pappas** received the S.B., S.M., and Ph.D. degrees in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, in 1979, 1982, and 1987, respectively.

He is currently with the Signal Processing Research Department at AT&T Bell Laboratories, Murray Hill, NJ. His research interests are in image processing and computer vision.

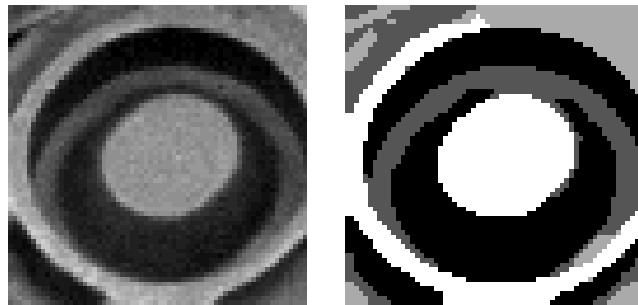


Figure 1



Figure 5





Figure 7

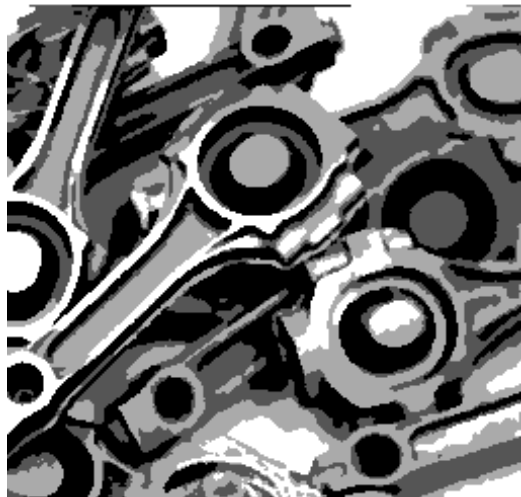
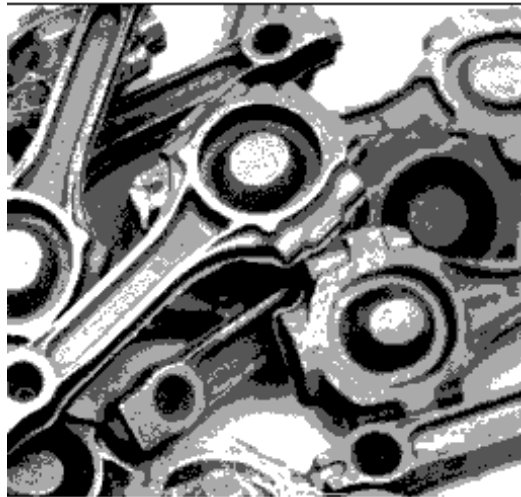
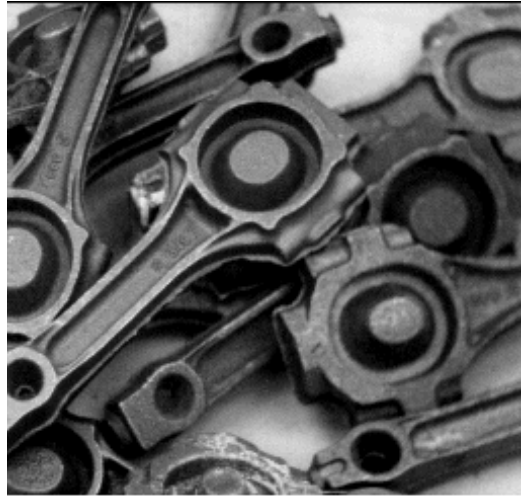


Figure 10

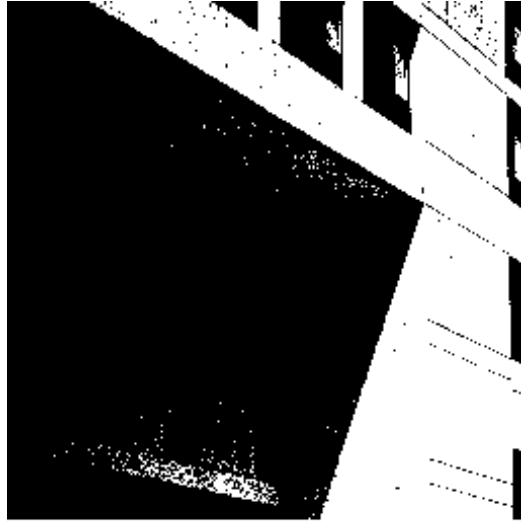


Figure 11

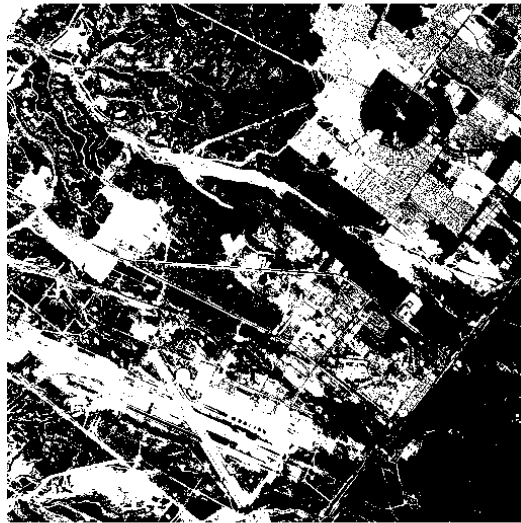


Figure 12

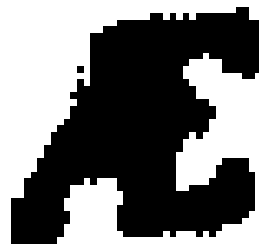
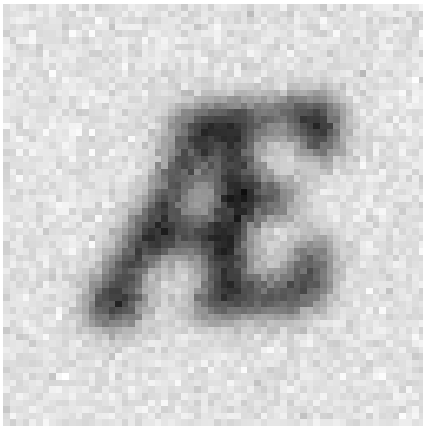


Figure 13

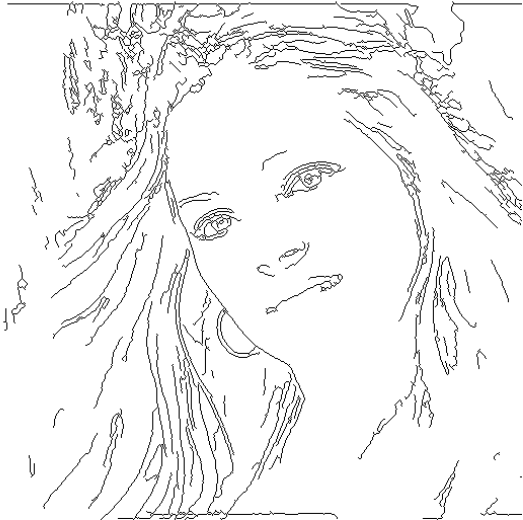


Figure 14



Figure 16

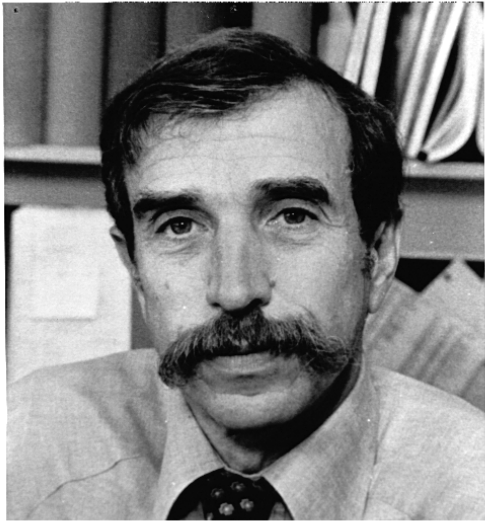
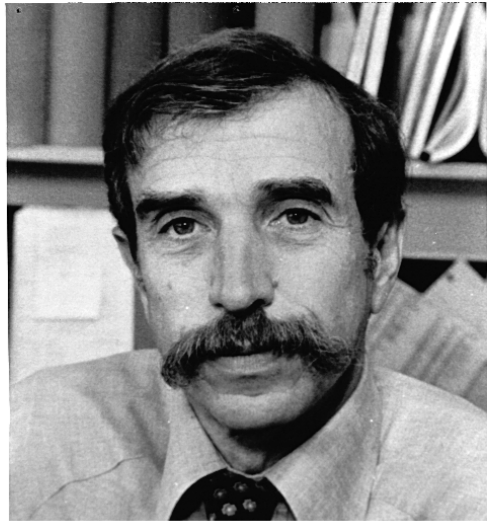


Figure 17