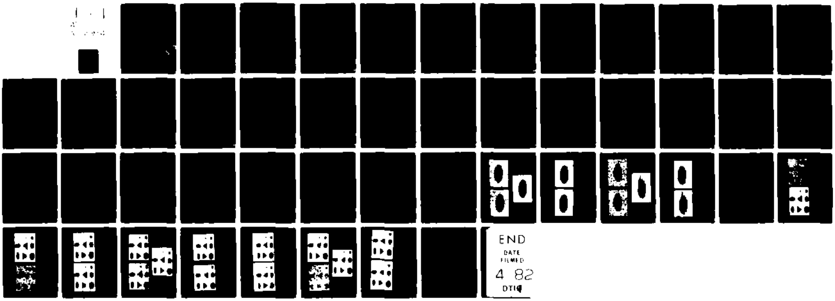
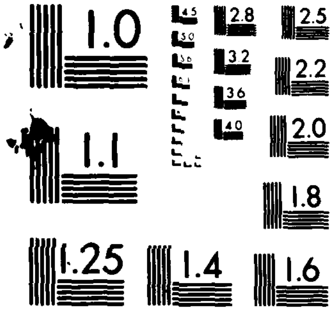
AD-A112 914 COLORADO STATE UNIV FORT COLLINS DEPT OF ELECTRICAL --ETC F/G 20/6
IMAGE SEGMENTATION USING SIMPLE MARKOV FIELD MODELS.(U)
OCT 81 F R HANSEN, H ELLIOTT N00014-82-K-0076
UNCLASSIFIED JU81-DELENG-1R NL



END
DATE
FILMED
4 82
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

Image Segmentation Using Simple Markov Field Models

by

F. R. Hansen[†]

H. Elliott[†]

Issued as

Colorado State University Technical Report #JU81-DELENG-1R

June 1981

October 1981

This work was supported by the Office of Naval Research under Grants N00014-75-C-0518 and N00014-82-K-0076.

[†] Department of Electrical Engineering, Colorado State University,
Fort Collins, CO 80523, Telephone: (303) 491-5306.

ABSTRACT

By modelling a picture as a two-state Markov field, MAP estimation technqiues are used to develop suboptimal but computationally tractable binary segmentation algorithms. The algorithms are shown to perform well at low signal to noise ratios, and analytical procedures are developed for estimating the Markov field transition probabilities. In addition, extensions of this approach to the multi-spectral and multi-region cases are discussed.

I. INTRODUCTION

The image segmentation process is a basic component of computer vision systems, and as such, has received considerable attention in the image processing literature. In this report we present a method for segmentation of two intensity level monochromatic pictures in the presence of high levels of additive noise. Although we concentrate on this limited class of images, our approach is extendable to a much more general class of images and this is briefly discussed in Section VI of the paper. In order to work with low signal to noise ratios, we have exploited the spatial dependence of pixel intensity by modelling the true underlying picture as a Markov field.

In the case of one dimensional digital signals the Markov field model reduces to a Markov chain. Markov chain models for stochastic processes have received considerable attention in the statistical literature [1], [2], and found extensive application in the control, communication, and signal processing fields [3]-[6]. Estimation and detection problems associated with random signals modelled by Markov chains can be formulated as likelihood maximization problems where one wants to maximize the joint likelihood of the data and the Markov state sequence. Maximization of this joint likelihood is completely equivalent to generating what is known as the maximum *a posteriori* probability estimate of the Markov chain [18]. In the one dimensional case, this leads to elegant and reasonably efficient dynamic programming algorithms for computing the estimate which maximizes the joint maximum *a posteriori* probability of the entire data string (digital signal) in a sequential manner. Unfortunately, this approach does not generalize in a natural way to the case of two dimensional signals such as digital images. As a result, this has somewhat hampered the use of MAP formulations in image processing. The few exceptions [7]-[12] are discussed more carefully below.

Recent work by Kaufman and co-workers, [7], and Therrien [8], [9] have also made use of Markov field models and MAP formulations. In [7] this approach was combined with reduced update Kalman filtering techniques for image enhancement, while in [8] and [9] it was combined with two-dimensional autoregressive texture models for texture based segmentation. However, the algorithm described in [7] and one of the algorithms described in [8] and [9] fail to exploit the true spatial dependence imposed by the model. In particular they don't attempt to maximize a joint likelihood of all the data but rather the individual likelihoods at points or in small regions within the image. This problem is partially overcome by a second multi-pass or iterative algorithm proposed in [8] and [9]. However, the relation between the estimate obtained using this approach and a true joint MAP estimate is unclear.

Cooper and Elliott [10]-[12] have applied MAP techniques to boundary estimation in noisy images as well. Although boundary estimation can be formulated as a one-dimensional signal estimation problem where the independent variable is arc length along the boundary, it is interesting to note that even in this case the two dimensional image data necessitated the use of a suboptimal algorithm.

In this report, we have taken a very simple Markov field model, and have developed an algorithm which approximates the behavior of an optimal sequential estimation algorithm. It makes use of two stages of dynamic programming. In the first stage a "generalized" dynamic programming algorithm is applied to each row of the image. This yields a set of candidate segmentations for each row. In the second stage, a final segmentation is "pieced" together from the candidate row segmentations using dynamic programming as well. The algorithm requires only a single pass over the image data, and the version for moderate signal to noise ratios can be performed in a highly parallel fashion. The version which works best at low signal to noise ratios requires a sequential raster processing of the image.

The paper is organized as follows. In Section II the problem of interest is formulated and the image model is discussed. In Section III the basic suboptimal algorithm is derived, while in Section IV, analysis tools are developed to allow estimation of the Markov field transition probabilities. Section V presents some modifications to the original algorithm to improve its performance at low signal to noise ratios, and extensions of this approach to a larger more interesting class of images is given in Section VI. Finally, some concluding remarks are made in Section VII.

II. MODELING ASSUMPTIONS AND PROBLEM FORMULATION

Let a digitized monochromatic picture be modeled as an $(N_1 x N_2)$ matrix B with components, b_{ij}, representing the intensity or grey level of pixel (i,j). Although this model will be generalized in Section VI, initially b_{ij} will be restricted to take on only one of two values, r_1 and r_2, $r_1 > r_2$. Thus the pixels in an image can be classified into two sets S_1 and S_2 where

$$S_1 = \{(i,j): b_{ij}=r_1\} \tag{1a}$$

$$S_2 = \{(i,j): b_{ij}=r_2\} \tag{1b}$$

It will be assumed that an observed image, G, represents a picture B corrupted by a stationary random noise field W so that G=B+W or equivalently, $g_{ij}=b_{ij}+w_{ij}$. The random variables w_{ij} are assumed to be independent, Gaussian, zero mean, and of known variance σ^2, denoted $w_{ij}\sim N(0,\sigma^2)$.

In this report we consider the problem of segmenting the image, or equivalently, generating estimates $\hat{S}_1$ and $\hat{S}_2$ of the sets S_1 and S_2. In the context of the model for B, this problem can be phrased as a two set classification problem where pixel (i,j) is assigned to $\hat{S}_1$ or $\hat{S}_2$ using the noisy observation g_{ij}. Since the conditional probability distributions

$$P(g_{ij}\leq t|(i,j)\epsilon S_k) = \int_{-\infty}^{t} (2\pi\sigma^2)^{-1/2}\exp(-(x-r_k)^2/2\sigma^2)dx \qquad k=1,2 \tag{2}$$

are assumed known, maximum likelihood classification techniques could be employed. This would result in a simple thresholding algorithm with pixel (i,j) assigned to $\hat{S}_1$ if $g_{ij}>A$ and to $\hat{S}_2$ if $g_{ij}\leq A$, where $A=(r_1+r_2)/2$. This method only works well if the signal to noise ratio $S/N \triangleq \Delta/\sigma > 2$, where $\Delta \triangleq r_1-r_2$.

In order to derive a more robust segmentation procedure, it is necessary to impose additional structure on B. Pixels in the set S_1 will be assumed to appear in connected subsets, or blobs, as illustrated in Figure 1. This structure can be modeled by a Markov field. Assuming the support for this field to be limited to a pixels four nearest neighbors, this Markov process can be characterized by the transition probabilities:

$$\begin{aligned} P(b_{ij}=r_k|B) &= P(b_{ij}=r_k|b_{i-1,j},b_{i+1,j},b_{i,j-1},b_{i,j+1}) \\ &\triangleq P_{ijk} \quad . \end{aligned} \tag{3}$$

This model is fairly general in nature, is direction invariant, and can be used to model a large class of clusters. Unfortunately, it is difficult to identify appropriate transition probabilities. In addition, optimal segmentation based upon such a model would be computationally impractical. Thus in the section to follow, we introduce a number of restrictions and approximations in order to arrive at a suboptimal but computational tractable segmentation algorithm based upon dynamic programming concepts.

III. SUBOPTIMAL SEGMENTATION

In order to motivate our suboptimal algorithm we first formulate an optimal algorithm which maximizes the joint likelihood of the image data, and the state of the Markov field assuming the transition probability structure given by (3).

An Optimal MAP Formulation

Let $\hat{S} = \{\hat{S}_1, \hat{S}_2\}$ represent an estimate of $S = \{S_1, S_2\}$, and let $\ell(\cdot)$ represent a log-likelihood function. The joint log-likelihood of $\hat{S}$ and the observed data G can be written as:

$$\ell(\hat{S},G) = \ell(G|\hat{S}) + \ell(\hat{S}) \quad . \tag{4}$$

The log-likelihood of the data conditioned on the segmentation $\hat{S}$ is

$$\ell(G|\hat{S}) = C - \sum_{(i,j)\epsilon\hat{S}_1} \frac{1}{2\sigma^2}(g_{ij}-r_1)^2 - \sum_{(i,j)\epsilon\hat{S}_2} \frac{1}{2\sigma^2}(g_{ij}-r_2)^2 \tag{5}$$

where C is a constant independent of the estimated segmentation $\hat{S}$. The log-likelihood of the estimated segmentation is

$$\ell(\hat{S}) = \sum_{(i,j)\epsilon\hat{S}_1} \ell n\, P_{ij1} + \sum_{(i,j)\epsilon\hat{S}_2} \ell n\, P_{ij2} \quad . \tag{6}$$

Combining (4), (5), and (6) yields

$$\ell(\hat{S},G) = \sum_{(i,j)\epsilon\hat{S}_1} (\ell n P_{ij1} - \frac{1}{2\sigma^2}(g_{ij}-r_1)^2) + \sum_{(i,j)\epsilon\hat{S}_2} (\ell n P_{ij2} - \frac{1}{2\sigma^2}(g_{ij}-r_2)^2) \tag{7}$$

For optimal segmentation (7) would have to be maximized over all possible segmentations $\hat{S}$. Even if appropriate transition probabilities could be identified, maximization of (7) is computationally impractical since it cannot be maximized in a sequential manner. In order to derive a computationally feasible algorithm $\ell(G,\hat{S})$ will be approximated by a function $\tilde{\ell}(G,\hat{S})$ which can be maximized by processing the observation matrix G in a sequential manner. It might be noted that the log likelihood (4) differs from the likelihood traditionally used for MAP estimation by only a term which is independent of $\hat{S}$. For the latter, one would use $\ell(\hat{s}/G) = \ell(G/\hat{S}) + \ell(\hat{S}) - \ell(G)$.

A Suboptimal Likelihood Function

To derive the approximation $\tilde{\ell}(G,\hat{S})$ we will restrict the Markovian dependence of b_{ij} to one side. In particular we will assume

$$P_{ijk} = P(b_{ij}=r_k|b_{i-1,j},b_{i,j-1}) \quad . \tag{8}$$

Use of (8) in (7) would still require sequencing through the image in two directions simultaneously for maximization. Since no efficient algorithm has been formulated for such a maximization an additional simplifying approximation will be made. In particular, we will approximate P_{ijk} by $\tilde{P}_{ijk}$ where

$$\tilde{P}_{ijk} = R_{ijk}\, C_{ijk} \tag{9a}$$

$$C_{ijk} = P(b_{ij} = r_k|b_{i-1,j}) \tag{9b}$$

$$R_{ijk} = P(b_{ij} = r_k|b_{i,j-1}) \tag{9c}$$

R_{ijk} represents a transition probability from the preceding pixel in the same row and C_{ijk} represents a transition probability from the preceding pixel in the same column. In the next section we will present some methods for estimating R_{ijk} and C_{ijk} from *a priori* knowledge of image properties. It might be noted, however, that if one starts with knowledge of the probabilities P_{ijk}, R_{ijk} and C_{ijk} can be calculated recursively. Using (9) we can define $\tilde{\ell}(G,\hat{S})$ by

$$\tilde{\ell}(G,\hat{S}) = \tilde{\ell}_R(G,\hat{S}) + \tilde{\ell}_C(\hat{S}) \tag{10a}$$

$$\tilde{\ell}_R(G,\hat{S}) = \sum_{(i,j)\epsilon\hat{S}_1} (\ell nR_{ij1} - \frac{1}{2\sigma^2}(g_{ij}-r_1)^2) + \sum_{(i,j)\epsilon\hat{S}_2} (\ell nR_{ij2} - \frac{1}{2\sigma^2}(g_{ij}-r_2)^2) \tag{10b}$$

$$\tilde{\ell}_C(\hat{S}) = \sum_{(i,j)\epsilon\hat{S}_1} \ell nC_{ij1} + \sum_{(i,j)\epsilon\hat{S}_2} \ell nC_{ij2} \tag{10c}$$

Thus by using (9) we have decomposed our likelihood into two components. The first, $\tilde{\ell}_R$, contains the image data and the row transition probabilities, while the second, $\tilde{\ell}_C$, contains only the column transition probabilities.

Suboptimal Algorithm for Maximization of $\tilde{\ell}(G,\hat{S})$

In order to use computationally efficient sequential algorithms for estimating $\hat{S}$ it will be necessary to maximize the approximate likelihood (10) in a suboptimal manner. In order to explain the algorithm it will be helpful to introduce some additional notation. Let

$G(i,J) \triangleq$ first J elements in ith row of G

$\hat{S}(i,J) = \{\hat{S}_1(i,J),\hat{S}_2(i,J)\}$

$\triangleq$ subset of $\hat{S}$ containing first J pixels in row i of image.

We can then write

$$\hat{S} = \bigcup_{i=1}^{N_1} \hat{S}(i,N_2)$$

and

$$\tilde{\ell}_R(G,\hat{S}) = \sum_{i=1}^{N_1} \tilde{\ell}_R(G(i,N_2),\hat{S}(i,N_2)) \qquad (11a)$$

$$\tilde{\ell}_R(G(i,N_2),\hat{S}(i,N_2)) = \sum_{(i,j)\in\hat{S}_1(i,N_2)} (\ln R_{ij1} - \frac{1}{2\sigma^2}(g_{ij}-r_1)^2)$$

$$+ \sum_{(i,j)\in\hat{S}_2(i,N_2)} (\ln R_{ij2} - \frac{1}{2\sigma^2}(g_{ij}-r_2)^2) \qquad (11b)$$

Our algorithm consists of two stages.

<u>Stage 1:</u> For each row i, $1 \le i \le N_1$, we calculate the M row sets $\hat{S}^m(i,N_2)$, $1 \le m \le M$, which yield the M largest values of $\tilde{\ell}_R(G(i,N_2),\hat{S}(i,N_2))$. Observe from (11) that

$$\tilde{\ell}_R(G(i,J),\hat{S}(i,J)) = \tilde{\ell}_R(G(i,J-1),\hat{S}(i,J-1)) + \hat{\ell}(g_{iJ}) \qquad (12)$$

where $\hat{\ell}(g_{iJ}) = \ell n R_{iJk} - \frac{1}{2\sigma^2}(g_{iJ}-r_k)^2$, for k=1 or k=2 depending upon whether pixel (i,J) is assigned to set $\hat{S}_1$ or $\hat{S}_2$. Recursion (12) in conjunction with the principle of optimality [13], imply the existence of a forward dynamic programming algorithm for finding the estimate $\hat{S}^1(i,N_2)$ which maximizes (11b). This type of dynamic programming algorithm is extended in a straight forward manner to generate the M most likely sets $\hat{S}^m(i,N_2)$, $1 \leq m \leq M$. A detailed description of the algorithm can be found in Appendix A.

Stage 2: Given the sets $\hat{S}^m(i,N_2)$, $1 \leq i \leq N_1$, $1 \leq m \leq M$, we calculate our complete estimate $\hat{S}$ as

$$\hat{S} = \bigcup_{i=1}^{N_1} \hat{S}^{\mu_i}(i,N_2)$$

where the sequence of sets $\{\hat{S}^{\mu_i}(i,N_2)\}_{i=1}^{N_1}$ maximize

$$\tilde{\ell}(G,\hat{S}) = \sum_{i=1}^{N_1} [\tilde{\ell}_R(G(i,N_2),\hat{S}^m(i,N_2)) + \tilde{\ell}_C(\hat{S}^m(i,N_2))] \qquad (13)$$

Define

$$\tilde{\ell}(G,\hat{S},I) = \sum_{i=1}^{I} \tilde{\ell}_R(G(i,N_2),\hat{S}^m(i,N_2)) + \tilde{\ell}_C(\hat{S}^m(i,N_2)) \qquad (14)$$

so that $\tilde{\ell}(G,\hat{S}) = \tilde{\ell}(G,\hat{S},N_1)$. By observing that

$$\tilde{\ell}(G,\hat{S},I) = \tilde{\ell}(G,\hat{S},I-1) + \tilde{\ell}_R(G(I,N_2),\hat{S}^m(I,N_2)) + \tilde{\ell}_C(\hat{S}^m(I,N_2)) \qquad (15)$$

We can again apply the principle of optimality to develop a forward dynamic programming algorithm for generating $\{\hat{S}^{\mu_i}(i,N_2)\}_{i=1}^{N_1}$. The details of this algorithm can also be found in Appendix A. At this point we make the following remarks with respect to this algorithm.

Remark 1: In developing this algorithm we have traded off optimality in order to reduce one two-dimensional maximization problem into (N_1+1) one-dimensional problems. To this point in our experimentation we have chosen

M such that $1 \leq M \leq 8$. Actually as M approaches 2^{N_2} the algorithm converges to an optimal algorithm in the sense that it generates an estimate $\hat{S}$ which maximizes (10).

Remark 2: This algorithm is attractive computationally since the time consuming Stage 1 is ideally suited for parallel implementation, where N_1 processors simultaneously process each row. Furthermore the amount of memory needed to store the Stage 1 results needed for execution of Stage 2 are modest. Each partition $\hat{S}^m(i,N_2)$ can be stored as a simple bit string of length N_2 where a one in position j implies pixel (i,j) is assigned to set $\hat{S}_1$ and a zero implies pixel (i,j) is assigned to set $\hat{S}_2$. Thus for each row of the image we need to store M bit strings of length N_2 and M numbers corresponding to the value of $\ell_R(G(i,N_2), \hat{S}^m(i,N_2))$.

Figures 2 and 3 give examples of the algorithms performance in finding an ellipse imbedded in noise such that $(r_1-r_2)/\sigma \triangleq \Delta/\sigma = 2$ and 1 respectively. The results are compared with a simple thresholding algorithm. For these simulations we chose M=1,4, and 8, and we used the transition probabilities

$$R_{ijk} = \begin{cases} T \text{ if pixel } (i,j-1)\varepsilon\hat{S}_k \\ 1-T \text{ otherwise} \end{cases} \tag{16a}$$

$$C_{ijk} = \begin{cases} T \text{ if pixel } (i-1,j)\varepsilon\hat{S}_k \\ 1-T \text{ otherwise} \end{cases} \tag{16b}$$

where T=.95. This choice for R_{ijk} and C_{ijk} imply both stationarity and symmetry in the i and j directions. Methods for choosing T are discussed in the following section. Observe that the algorithm as given above, is subject to burst type errors where strings of pixels are misclassified.

This is a common attribute of algorithms which estimate Markovian processes using maximum likeiihood techniques. Fortunately these type errors are easily compensated for using simple post-processing algorithms. This and other improvements to the basic algorithm are discussed in detail in Section V.

IV. ALGORITHM ANALYSIS FOR PARAMETER CALCULATIONS

In this section a simplified error analysis is presented. Using techniques similar to those used in [4], we calculate the probability that a given incorrect row segmentation is more likely than the underlying true row segmentation. This probability is then used to obtain an estimate of the paramater T in terms of certain *a priori* information about the structure of B. Some examples of the algorithms performance with various values of T will then be presented.

Simplified Error Analysis

We make the same assumptions about the transition probabilities C_{ijk} and R_{ijk} that were used for the simulations described in Section III. In particular, we assume stationarity and symmetry so that (16) holds. Let $\hat{S}(i,N_2) = \{\hat{S}_1(i,N_2),\hat{S}_2(i,N_2)\}$ be an incorrect estimate of the true segmentation $S(i,N_2) = \{S_1(i,N_2),S_2(i,N_2)\}$ of row i. Consider the probability $P_E(i)$, that $\hat{S}(i,N_2)$ is more likely than $S(i,N_2)$. Using the stage 1 likelihood function, $\tilde{\ell}_R(\cdot.\cdot)$, we obtain

$$P_E(i) = P(\tilde{\ell}_R G(i,N_2),\hat{S}(i,N_2)) > \tilde{\ell}_R(G(i,N_2),\hat{S}(i,N_2))) \tag{17}$$
$$\triangleq P(Q(i) > 0)$$

where

$$Q(i) \triangleq \tilde{\ell}_R(G(i,N_2),\hat{S}(i,N_2)) - \tilde{\ell}_R(G(i,N_2),S(i,N_2)) \tag{18}$$

Using (11b) and expanding the quadratic terms, Q(i) can be rewritten as

$$\begin{aligned} Q(i) = &\sum_{(i,j)\in\hat{S}_1(i,N_2)} \left(\ell n\, R_{ij1} + \frac{1}{2\sigma^2}(2g_{ij}r_1 - r_1^2)\right) \\ &+ \sum_{(i,j)\in\hat{S}_2(i,N_2)} \left(\ell n\, R_{ij2} + \frac{1}{2\sigma^2}(2g_{ij}r_2 - r_2^2)\right) \\ &- \sum_{(i,j)\in S_1(i,N_2)} \left(\ell n\, R_{ij1} + \frac{1}{2\sigma^2}(2g_{ij}r_1 - r_1^2)\right) \\ &- \sum_{(i,j)\in S_2(i,N_2)} \left(\ell n\, R_{ij2} + \frac{1}{2\sigma^2}(2g_{ij}r_2 - r_2^2)\right) \end{aligned}$$

In these expressions R_{ijk} will take the value T if pixels (i,j-1) and (i,j) are placed into the same set, and will take the value (1-T) otherwise. Let t(i) be the number of times pixels (i,j-1) and (i,j) are classified differently in $S(i,N_2)$, and let $\hat{t}(i)$ be defined analogously for $\hat{S}(i,N_2)$. If $\bar{S}_k(i,N_2)$ is defined as the set of pixels contained in $\hat{S}_k(i,N_2)$ and not in $S_k(i,N_2)$, i.e., $\bar{S}_k(i,N_2) = \hat{S}_k(i,N_2) - S_k(i,N_2)$ then Q(i) can be simplified to

$$Q(i) = \frac{1}{2\sigma^2} \Big[\sum_{(i,j)\varepsilon\bar{S}_2(i,N_2)} (r_1^2-r_2^2-2(r_1-r_2)g_{ij}) + \sum_{(i,j)\varepsilon\bar{S}_1(i,N_2)} (r_2^2-r_1^2-2(r_2-r_1)g_{ij})\Big] + (t(i) - \hat{t}(i))\ln\frac{T}{1-T} \qquad (19)$$

for $\hat{S}(i,N_2)$. Let $\bar{n}_k(i)$ be the number of pixels in the set $\bar{S}_k(i,N_2)$. Then, conditioned on the knowledge that $S(i,N_2)$ is correct

$$Q(i) \sim N\left(\frac{-\Delta^2}{2\sigma^2}(\bar{n}_1(i) + \bar{n}_2(i)) + (t(i) - \hat{t}(i))\ell n(T/1-T),\ (\frac{\Delta}{\sigma})^2(\bar{n}_1(i) + \bar{n}_2(i))\right) \tag{20}$$

Thus

$$P_E(i) = \int_{\alpha(i)}^{\infty} (2\pi)^{-1/2}\exp(-\frac{s^2}{2})ds \tag{21a}$$

where

$$\alpha(i) = \frac{(\bar{n}_1(i)+\bar{n}_2(i))^{1/2}}{2}(\frac{\Delta}{\sigma}) - (\frac{\sigma}{\Delta})\frac{t(i)-\hat{t}(i)}{(\bar{n}_1(i)+\bar{n}_2(i))^{1/2}}\ell n(\frac{T}{1-T}) \tag{21b}$$

As expected, $P_E(i)$ goes to zero as the signal to noise ratio, Δ/σ, goes to infinity. The contribution of the transition probabilities to $P_E(i)$ diminishes with increasing signal to noise ratio, and the transition probability term only influences $P_E(i)$ when $t(i)\neq\hat{t}(i)$, or when the number of state changes in $S(i,N_2)$ is different than the number in $\hat{S}(i,N_2)$.

Markov Parameter Determination

Using (20) and (21), bounds on the parameter T can be determined. These bounds can be arrived at by looking at two limiting cases. One case is when T is very large so that the probability of changing states is low. Here it would be expected that an object, a connected set of pixels in either S_1 or S_2, might be deleted from the estimated segmentation, $\hat{S}$, due to the low probability of any state transistions from S_1 to S_2 or S_2 to S_1. The other extreme is T very small which makes the probability of changing states larger. In this case a number of extraneous transitions might be included in $\hat{S}$. This situation produces what might be called "false objects", and will be considered first.

For the first case, let $\hat{S}(i,N_2)$ be such that all pixels in a connected set, Γ_1, in row i are misclassified. Furthermore, it will be assumed that the set Γ_1 is such that the misclassification causes $\hat{S}(i,N_2)$ to have 2 state transitions more than $S(i,N_2)$. An illustration of such an occurrence appears in Figure 4a.

If the set Γ_1 is n_1 pixels long, then the probability that $\hat{S}(i,N_2)$ will be more likely than $S(i,N_2)$ is given by (17) with

$$\alpha(i) = \frac{(n_1)^{1/2}}{2} (\frac{\Delta}{\sigma}) + (\frac{\sigma}{\Delta}) \frac{2}{n_1^{1/2}} \ell n(T/1-T) \tag{22}$$

Thus to assure that $P_E(i)$ is small, i.e. $P_E(i) < \varepsilon$, T should be picked such that $\alpha(i)$ is greater than or equal to some value $\beta(\varepsilon)$. From (22) this is equivalent to requiring

$$T \geq \exp(\beta(\varepsilon)x-x^2) \; (1+\exp(\beta(\varepsilon)x-x^2)) \tag{23a}$$

where

$$x = \frac{n_1^{\frac{1}{2}}}{2} |\frac{\Delta}{\sigma}| \tag{23b}$$

Note that the left hand side of this inequality is maximized at $x = \beta(\varepsilon)/2$, thus

$$T \geq \exp(\frac{\beta(\varepsilon)^2}{4})/(1+\exp(\frac{\beta(\varepsilon)^2}{4})) \tag{24}$$

will guarantee that

$$P_E(i) \leq \int_{\beta(\varepsilon)}^{\infty} (2\pi)^{-1/2} \exp(-s^2/2)ds = \varepsilon \tag{25}$$

For example, if we set $\varepsilon = .01$ then $\beta(\varepsilon) = 2.33$. This will require that $T \geq .80$ which is quite reasonable.

If (22) is minimized with respect to n_1, we see that given

$(\frac{\Delta}{\sigma})$ and T, the most probable length of a false object is approximately

$$n^* = 4\ell n(T/1-T)(\frac{\sigma}{\Delta})^2 \qquad (26)$$

Also, it should be noted that the event of the row segmentation $\hat{S}(i,N_2)$ being more likely than $S(i,N_2)$ is dependent only on the pixels in Γ_1. Therefore one estimate for the number of possible independent sites for such a set to occur might be $(N_1 N_2)/(n^*+1)$.[†] Since the corrupting noise was assumed white, the number of such sets, Γ_1, in the segmentation, S, can be similarly estimated as

$$\eta_1 = N_1 N_2 P_E / (n^*+1) \qquad (27)$$

In practice, however, this estimate seems to be consistently low by a factor of approximately four.

Thus, given an upper bound on the probability of $\hat{S}$ containing a false object in any given position, a lower bound on T can be obtained. Once T is chosen both the most probable length of, and the expected number of such sets can be estimated by n^* and $4\eta_1$ respectively.

In order to obtain an upper bound on T the second case must be considered. Here, it is assumed that an entire connected set of pixels, Γ_2, in $S_k(i,N_2)$, k=1,2, has been falsely classified in $\hat{S}_k(i,N_2)$. By entire, we mean that the pixels immediately to either side of Γ_2 are not elements of $S_k(i,N_2)$. In this case $\hat{S}(i,N_2)$ has two fewer state transitions than $S(i,N_2)$. This case is illustrated in Figure 4b.

Again, it is assumed that $\hat{S}(i,N_2)$ includes no other errors and that Γ_2 is n_2 pixels long. In this case the probability of an error $P_E(i)$,

[†] n^*+1 is used rather than n^* to insure each connected set is separated from the others by at least one pixel. Thus, each "false object" will contribute 2 extraneous transistions to $\hat{t}(i)$.

as defined by (17) and (18), is given by the integral (21a) where

$$\alpha(i) = \frac{n_2^{\frac{1}{2}}}{2} (\frac{\Delta}{\sigma}) - (\frac{\sigma}{\Delta}) \frac{2}{n_2^{\frac{1}{2}}} \ln(\frac{T}{1-T}) \tag{28}$$

If we use similar arguments to those of the previous case, in order for $P_E(i) \leq \varepsilon$, we would require $\alpha(i) \geq \beta(\varepsilon)$. Using (28) in this case yield the upper bound on T:

$$T \leq \exp(\frac{n_2}{4} (\frac{\Delta}{\sigma})^2 - \beta(\varepsilon) \frac{n_2^{\frac{1}{2}}}{2} (\frac{\Delta}{\sigma}))/1+\exp(\frac{n_2}{4}(\frac{\Delta}{\sigma})^2 - \beta(\varepsilon) \frac{n_2^{\frac{1}{2}}}{2} (\frac{\Delta}{\sigma})) \tag{29}$$

Minimization of (29) with respect to n_2 and $(\frac{\Delta}{\sigma})$ yields

$$T \leq \exp(- \frac{\beta(\varepsilon)^2}{4})/(1+\exp(- \frac{\beta(\varepsilon)^2}{4})) \quad . \tag{30}$$

This implies $T \leq .5$ which is inconsistent with the constraints imposed on T by the lower bound (24) derived from analysis of case 1. More importantly, it is inconsistent with our basic picture assumption of relatively large region clusters.

This problem can be overcome by assuming specific values for the signal to noise ratio, (Δ/σ), and for the length n_2 of the misclassified pixel string Γ_2. In this case if T is chosen to satisfy (29) for a specific ε, then any string of length greater than n_2 would have lower error probability. Tables 1 and 2 give the upperbound on T as defined by (29) for values of ε ranging from .16 to .02 and values on n_2 ranging from 4 to 40. In Table 1 $\Delta/\sigma = 1$ and in Table 2 $\Delta/\sigma = 2$.

Figure 5a shows a test image containing ellipses, triangles and circles with different orientations and sizes, and Figure 5b shows this same image with additive Gaussian noise such that $\Delta/\sigma = 1$. The results of applying our algorithm with T=.5, .8, .9, and .95 are shown in Figures 5c-5f. Note that for small values of T many additional state transitions occur and false objects appear. On the other hand, large values of T tend to suppress state transitions causing case 2 situations. Figures 6a-6c show the algorithm performance at $\Delta/\sigma = 2$ and $T = .95$ and $.5$. With $T = .5$ our algorithm reduces to maximum likelihood thresholding as discussed in Section II.

n_2 ε	.16	.10	.05	.02
4	.50	.43	.34	.26
6	.57	.48	.37	.27
8	.64	.54	.42	.29
10	.71	.61	.47	.32
12	.78	.68	.54	.37
14	.84	.75	.60	.42
15	.86	.78	.64	.45
20	.94	.89	.79	.60
25	.98	.95	.89	.75
30	.99	98	.95	.87
35	.997	.99	.98	.94
40	.999	.997	.99	.97

Table 1. Upperbounds on T determined from Equation (29) with $\Delta/\sigma = 1$.

n_2 ε	.16	.10	.05	.02
1	.5	.43	.34	.26
2	.64	.54	.42	.29
3	.78	.68	.54	.37
4	.88	.81	.67	.48
6	.97	.94	.88	.73
8	.99	.99	.97	.91
10	.999	.997	.99	.97

Table 2. Upperbounds on T determined from Equation (29) with $\Delta/\sigma = 2$.

V. MODIFICATIONS

As is apparent from our previous examples, the basic algorithm as described in Section III performs well for signal to noise ratios of 1.5 or greater. However, its performance is disappointing at signal to noise ratios near or below 1.0. An obvious approach to improving performance would be to increase M, the number of row alternatives stored during stage 1, to a value greater than M=8. Unfortunately, as is discussed below, this is impractical. Consequently, we present two alternative, but more practical modifications, which when applied jointly yield reasonable performance at signal to noise ratios of 1 or below.

Effects of Increasing M

As can be seen in Figures 2 and 3, as M increases algorithm performance does improve. However, certain row estimates experience little or no improvement. This is especially true in the case $\Delta/\sigma = 1$. To understand the effect of increasing M one must realize that stage 2 of the algorithm can not generate a reasonable picture segmentation if the true row segmentation, or a reasonable estimate, is not contained in the set of candidate row estimates generated during stage 1. With this in mind we considered three troublesome rows in Figure 3c, namely rows 11, 27, and 32, extracted them from the image, and applied the stage 1 algorithm to them with M=64. The 20th and 12th most likely alternatives for rows 11 and 13 respectively did yield an improved estimate of their true segmentations. However, no such alternative appeared for row 27. This implies that improved performance is only possible if M is increased substantially. This in turn would lead to impractical time and storage requirements. As a result we have considered two other alternative modifications. The first is a simple post-filtering operation. The second involves a modification of stage 1 of the algorithm.

Post-Filtering

The errors incurred by the algorithm are generally "burst" type errors where a sequence of pixels in a given row are misclassified. Thus, one way to remove the effects of such errors is to post-filter the resulting binary image representing the segmentation. There are many types of filters which could be used, and we show the results of applying one simple type. We reassign the state of a pixel if the majority of the pixels in a 5x1 window centered on the pixel are in a different state. This is just a special case of a median filter. The result of applying this filter to the binary segmentation shown in Figure 5f is given in Figure 7a, while Figure 7b shows the results of applying the filter to Figure 6c. For purposes of comparison we have also filtered the segmentations shown in 5c and 6b which were obtained by simple thresholding. Since the errors in these images do not exhibit any directional bias, we used a (3x3) window. As can be seen by comparing Figures 7c and 7d to 5c and 6b, respectively, the approximate MAP algorithm produces significant improvement at $\Delta/\sigma = 1$ and only minor improvement at $\Delta/\sigma = 2$. Thus the additional complexity of the MAP algorithm can only be justified when working with signal to noise ratios lower than 2.

Non-Parallel Algorithm

Although post-filtering produces significant improvement in performance when compared with the output of the basic algorithm, one would still prefer improving the performance of the basic algorithm before post-processing. In view of the discussion above, our goal is to improve the list of candidate row estimates generated by stage 1. This can be done by using information from the previously processed rows. Unfortunately, to do this we must remove the parallelism from the algorithm and process the rows sequentially.

In this modified approach it makes more sense to interweave stages 1 and 2 of the algorithm. That is, after processing row i with a modified version of the stage 1 algorithm to obtain the most likely candidate segmentations, we then perform one step in the stage 2 dynamic programming algorithm. At this point we have M distinct binary segmentations stored for the sub-image consisting of the first i rows.

The first stage algorithm is modified to make use of the dynamic programming solution for the previous i-1 rows as well. In particular we now find the M candidate estimates which yield the largest values of the modified row likelihood

$$\tilde{\ell}_{MR}(G(i,N_2),\hat{S}(i,N_2)) = \tilde{\ell}_R(G(i,N_2)\hat{S}(i,N_2)) + \sum_{(i,j)\epsilon\hat{S}_1(i,N_2)} \ell n\, H_{ij} + \sum_{(i,j)\epsilon\hat{S}_2(i,N_2)} \ell n\, H_{ij2} \qquad (31)$$

where $\tilde{\ell}_R(\cdot)$ is given by (11b), and

$$H_{ij} = \begin{cases} T_R & \text{if } \hat{b}_{ij} = b^*_{i-1,j} \\ 1-T_R & \text{otherwise} \end{cases} \qquad (32)$$

The term $\hat{b}_{ij}$ is the state estimate for pixel (i,j) and $b^*_{i-1,j}$ is the most likely state estimate for pixel (i-1,j) after running both stages of the algorithm on the first i-1 rows of the image. In this case we have incorporated information about our segmentation of the previous rows in making decisions regarding new rows. In particular we penalize state changes in the vertical direction as well as the horizontal direction when performing our initial row segmentations. Observe that the addition of the H_{ij} terms in (31) does not change its recursive nature and hence dynamic programming can still be used to maximize (31).

Finally we point out that before performing the stage 2 algorithm on a given row the H_{ij} terms are subtracted out of the row likelihoods, $\tilde{\ell}_{MR}(\cdot)$, reducing it back to the form of $\tilde{\ell}_R(\cdot)$ given in (11b). This assures that the final image likelihood is consistent with our Markov field model.

Estimating T and T_R

In order to implement this modified algorithm we must estimate appropriate values for T and T_R. The approach taken is similar to that used for the original algorithm and the details can be found in [11]. However, we briefly outline the necessary changes.

To estimate T alone we considered two types of row errors. To simultaneously estimate the two parameters T and T_R we consider three error types. Case 1 is identical to the first "false object" case used in the earlier analysis. Case 2 corresponds to the probability of missing the top row of an object while Case 3 consists of the probability of a protusion of one pixel width extending from an object. If P_{E1}, P_{E2}, and P_{E3} are defined as the error probabilities for each of these cases reasonable values for T and T_R can be determined by minimizing

$$P^* = \gamma_1 P_{E1} + \gamma_2 P_{E2} + \gamma_3 P_{E3} \quad . \tag{33}$$

The positive parameters γ_1, γ_2, and γ_3 are adjusted depending upon what type of error is more reasonable in a given application. In our case we were most concerned with eliminating "false objects."

Thus we used γ_1=100 and $\gamma_2=\gamma_3 = 1$. For $\Delta/\sigma = 0.8$, minimization of (33) yielded T=.83, T_R = .55, $P_{E1} = 6.6\text{x}10^{-4}$, P_{E2} = .81 and P_{E3} = .07. Interestingly, as one would expect with this algorithm, P_{E2} is large. That is there is a high probability of not estimating the first row of objects. Figure 8a shows the image of Figure 5a with additive noise such that Δ/σ = .8 while Figure 8b shows the result of applying the modified algorithm to this image with T and T_R as determined above. Figure 8c is the post filtered version of Figure 8b. Figure 9a shows the result of applying the modified algorithm on image 5b which has Δ/σ = 1. In this case using the above analysis procedures we chose T = .83, T_R = .6. The corresponding error probabilities were $P_{E1} = 3.4\text{x}10^{-4}$, P_{E2} = .76 and P_{E3} = .02. Figure 9b is the post-filtered version of 9a.

VI. EXTENSIONS

Although in this report we have concentrated on the simple uniform intensity binary segmentation problem we point out that the approach is easily extended.

First, if one were interested in the colored or multi-spectral case, one could model the image as a set of intensity matrices $\{G_\ell = B_\ell + W_\ell\}_{\ell=1}^{L}$ where each matrix corresponded to the intensity level of a primary color or of a given spectral band. In this case a region would be characterized by the set of intensities $\{r_k^\ell\}_{\ell=1}^{L}$.

Next observe that by increasing the number of allowable Markov states, the multi-region case can be handled. If the image had K regions then a segmentation would be characterized by $\hat{S} = \bigcup_{k=1}^{K} \hat{S}_k$.

Combining these two extensions, leads to the following generalization of (5)

$$\ell(\hat{S},\{G_\ell\}) = \ell(\hat{S}) + \ell(\{G_\ell\}|\hat{S}) \tag{34}$$

$$\ell(\{G_\ell\}|\hat{S}) = C - \sum_{\ell=1}^{L} \left[\sum_{k=1}^{K} \left(\sum_{(i,j)\varepsilon\hat{S}_k} \frac{1}{2\sigma^2}(g_{ij}-r_k^\ell)^2 \right) \right] \tag{35}$$

$$\ell(\hat{S}) = \sum_{k=1}^{K} \sum_{(i,j)\varepsilon\hat{S}_k} P_{ijk} \tag{36}$$

Using (34)-(36) estimation algorithms similar to those described earlier can be developed.

Finally we point out that this approach can be even further extended to consider textured images where region textures are modelled stochastically using probability density functions or dynamical models as in [8], [9]. In this case the state space of the Markov field must be modified. That is, rather than characterize a region by a simple feature such as intensity, one could define it to be a parameter vector associated with the texture model.

VII. CONCLUDING REMARKS

This paper presented a new segmentation algorithm based upon a simple Markov field model for an image and MAP estimation techniques. The two-dimensional nature of the image forced us to consider a suboptimal algorithm for actual segmentation. We also presented some analytical techniques for estimating key algorithm parameters. Some areas for future work include more careful study of the extensions outlined in the previous section. In addition it would be of interest to explore other approximations to the true MAP estimate and compare performance of these alternative approaches both experimentally and analytically.

REFERENCES

[1] J. L. Doob, Stochastic Processes, John Wiley, New York, P. 153.

[2] K. L. Chung, A Course in Probability Theory, Academic Press, New York, 1974 (Second Edition).

[3] H. Kushner, Introduction to Stochastic Control, Holt, Rinehart and Winston, New York, 1971.

[4] G. D. Forney, "Maximum Likelihood Sequence Estimation of Digital Sequences in the Presence of Intersymbol Interference," IEEE Trans. on Info. Theory, Vol. IT-18, May 1973, pp. 363-377.

[5] A. J. Viterbi, "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm" IEEE Trans. on Info. Theory, Vol. IT-13, April 1967, pp. 363-377.

[6] L. L. Scharf, D. D. Cox, C. J. Masreliez, "Modulo-2π Phase Sequence Estimation," IEEE Trans. on Info. Theory, Vol. IT-26, Sept. 1980, pp. 615-620.

[7] H. Kaufman, J. W. Woods, V. K. Ingle, R. Mediqvilla, and A. Radpour, "Recursive Estimation: A Multiple Model Approach," Proc. 18th Conf. on Dec. and Control, Fort Lauderdale, Dec. 1979.

[8] C. W. Therrien, "Linear Filtering Models for Texture Classification and Segmentation," Proc. of 5th Int. Conf. on Pat. Rec., Miami, Dec. 1980.

[9] C. W. Therrien, "Linear Filtering Models for Terrain Image Segmentation," MIT Lincoln Laboratory Tech. Rep. #552, February 1981.

[10] H. Elliott, D. B. Cooper, F. Cohen, P. Symosek, "Implementation, Interpretation and Analysis of a Suboptimal Boundary Finding Algorithm," Col. St. Univ. Tech. Rep. #JA80-DELENG-1, Jan. 1980 (to appear IEEE Trans. on PAMI).

[11] H. Elliott, L. Srinivasan, "An Application of Dynamic Programming to Sequential Boundary Estimation," Col. St. Univ. Tech. Rep. #AP81-DELENG-1, April 1981 (to appear Computer Graphics and Image Processing).

[12] D. B. Cooper, H. Elliott, F. Cohen, L. Reiss, P. Symosek, "Stochastic Boundary Estimation and Object Recognition," Computer Graphics and Image Processing, April 1980, pp. 326-355.

[13] R. Bellman, Dynamic Programming, Princeton University Press, Princeton, 1957.

[14] J. Ravin, "Decision Making in Markov Chains Applied to the Problem of Pattern Recognition," IEEE Trans. on Info. Theory, Vol. IT-13, Oct. 1967, pp. 536-551.

[15] L. L. Scharf and H. Elliott, "Aspects of Dynamic Programming in Signal and Image Processing," to appear IEEE Trans. on Aut. Control, Bellman Issue, Oct. 1981.

[16] H. Elliott and F. R. Hansen, "An Application of Adaptive Algorithms to Image Processing, Proceedings of 19th Conference on Decision and Control, Albuquerque, 1980.

[17] F. R. Hansen, "Application of Markov Field Models to the Design and Analysis of Image Segmentation Algorithms," M.S. Thesis, Colorado State University, July 1981.

[18] A. P. Sage, J. L. Melsa, Estimation Theory with Application to Communications and Control, McGraw-Hill, New York, 1971.

APPENDIX A

This appendix describes the dynamic programming techniques necessary for implementation of the Stage 1 and Stage 2 algorithms discussed in Section III. We first present a basic forward dynamic programming algorithm, of the form popularized by Bellman [13], for segmenting a row into regions, $\hat{S}^k$, characterized by region intensities r_k, and the Gaussian noise model. Modification of this algorithm for generating the M most likely segmentation (the Stage 1 algorithm) are then presented. Finally, the Stage 2 algorithm is derived by simply changing a few of the definitions introduced for row segmentation.

<u>Forward Dynamic Programming for Segmentation of the ith Row of an Image</u>

Preliminary Definitions:

$$D_{kj} \triangleq -\frac{1}{2\sigma^2}(g_{ij}-r_k)^2$$

$$\pi_{kj} \triangleq \begin{matrix} \ell nT & \text{if } k=j \\ \ell n(1-T) & \text{if } k\neq j \end{matrix}$$

$$K \triangleq 2$$

$$N \triangleq N_2$$

Step 1: Set $c = 1$

$$\ell_k^c = D_k(g_{i1}) \quad k=1,\ldots,K$$

Step 2: Set $c = c+1$

$$\ell_k^c = \max_{1\leq j\leq K} (\ell_j^{c-1} + D_{kc} + \pi_{jk}) \text{ , } k=1,2,\ldots K$$

$$\gamma_k^{c-1} = \text{maximizing value of } j,\ k=1,\ldots,K$$

Step 3: If $c<N$ go to Step 2

Step 4: Set $\tilde{\ell}_R(G(1,N_2),\hat{S}(i,N_2)) = \max_{1\leq j\leq K} \ell_j^{N_2}$

$$\rho = \text{maximizing value of } j$$

```
        Assign pixel $(i,N_2)$ to set $\hat{S}_\rho(i,N_2)$
Step 5: $c = c-1$
        $\rho = \gamma_\rho^c$
        Assign pixel $(i,c)$ to set $\hat{S}_\rho(i,N_2)$
Step 6: If $c>1$ go to 5
Step 7: Stop.
```

The Stage 1 Algorithm

To generate the M most likely segmentations modify steps 2, 4, and 5 above to

```
Step 2': $\ell_k^{c,m} \underset{\substack{1\le j\le K\\ 1\le\mu\le M}}{\text{m-max}}(\ell_j^{c-1,\mu} + \pi_{jk} + D_{kj})$, $k=1,\dots,K$, $M=1,\dots,M$
         $\gamma_k^{c-1,m}$ = maximizing value of $j$ ,  $k=1,\dots,K$, $M=1,\dots,M$
         m-max$(\cdot)$ = mth largest .
Step 4': $\tilde{\ell}_R(G(i,N_2)\hat{S}^m(i,N_2) = \underset{\substack{1\le j\le K\\ 1\le\mu\le M}}{\text{m-max}}(\ell_j^{N_2,\mu})$ , $m=1,\dots,M$
         $\rho^m$ = maximizing value of $j$ ,  $m=1,\dots,M$
         Assign pixel $(i,N_2)$ to set $\hat{S}^m_{\rho^m}(i,N_2)$ ,  $m=1,\dots,M$
Step 5': $c=c-1$
         $\rho^m = \gamma^{c,m}_{\rho^m}$ , $m=1,2,\dots,M$
         Assign pixel $(i,c)$ to set $\hat{S}^m_{\rho^m}(i,N_2)$ , $m=1,\dots,M$ .
```

The Stage 2 Algorithm

The Stage 2 algorithm is a simple forward dynamic programming algorithm, and as such, can be obtained from the basic row segmentation algorithm by redefining the following parameters:

$$D_{kj} = \tilde{\ell}_R(G(j,N_2),\hat{S}^k(j,N_2))$$

$$\pi_{kj} = \tilde{\ell}_c(\hat{S}^k(j,N_2))$$

$K = 8$

$N = N_1$

In addition steps 4 and 5 should be redefined as:

Step 4": $\tilde{\ell}(G,\hat{S}) = \max\limits_{1\leq j\leq K} \ell_j^N$

ρ=maximizing value of j

Assign row segmentation $\hat{S}^{\rho}(N,N_2)$ to $\hat{S}$

Step 5": $c = c-1$

$\rho = \gamma_p^c$

Assign $\hat{S}^{\rho}(c,N_2)$ to $\hat{S}$

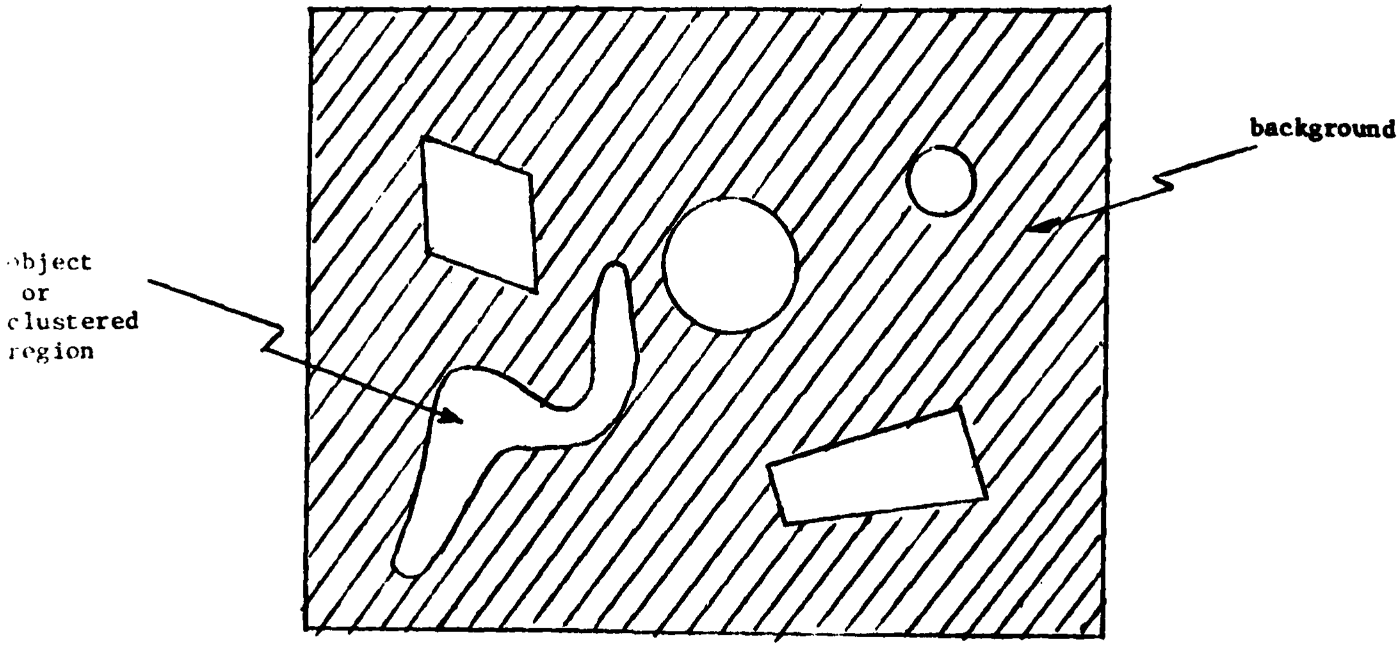


Figure 1. Examples of clustered regions.

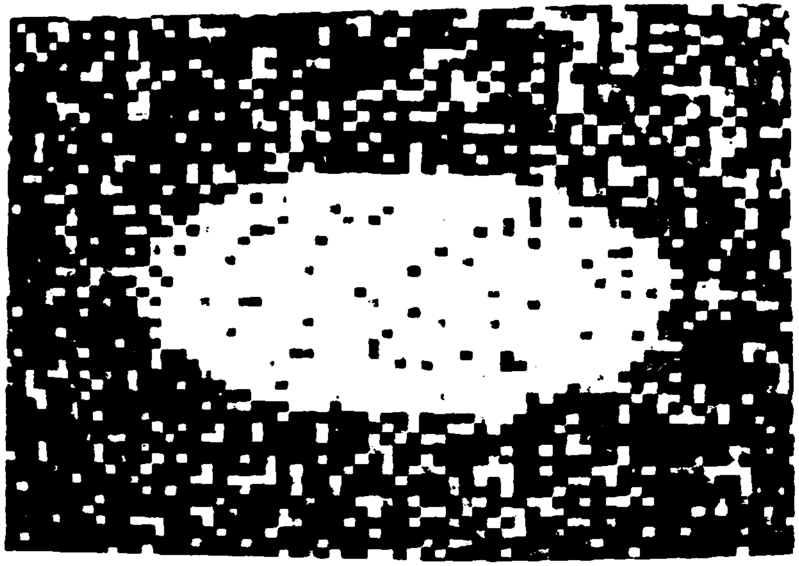


Figure 2a. 64x64 ellipse in additive Gaussian noise such that $\Delta/\sigma = 2$.

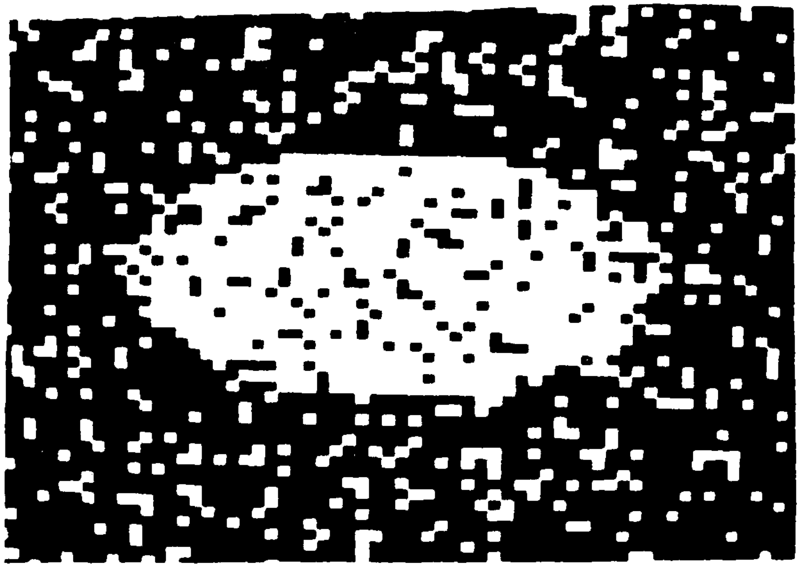


Figure 2b. Maximum likelihood threshold segmentation.

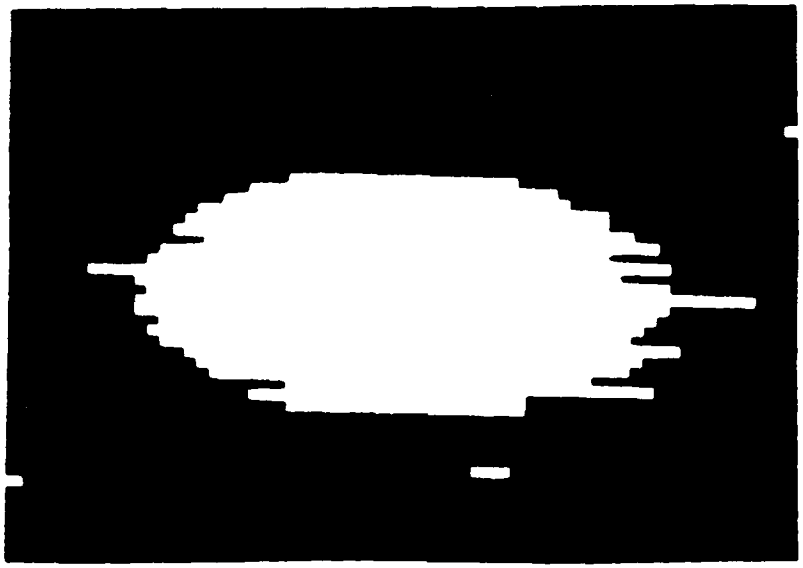


Figure 2c. MAP segmentation with T=.95 and M=4.

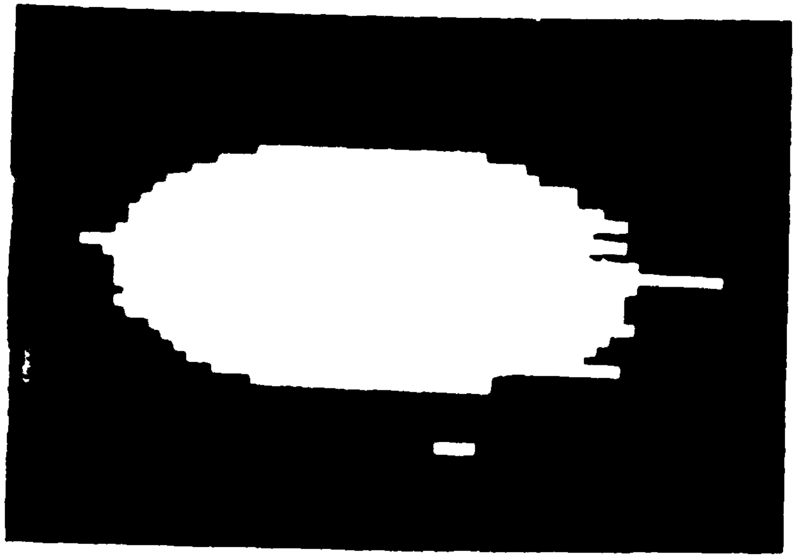


Figure 2d. MAP segmentation with T=.95 and M=4.

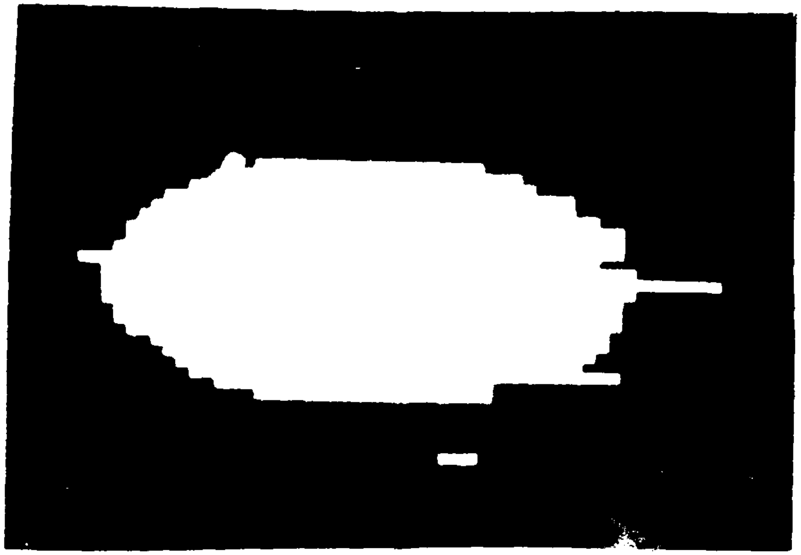


Figure 2e. MAP segmentation with T=.95 and M=8.

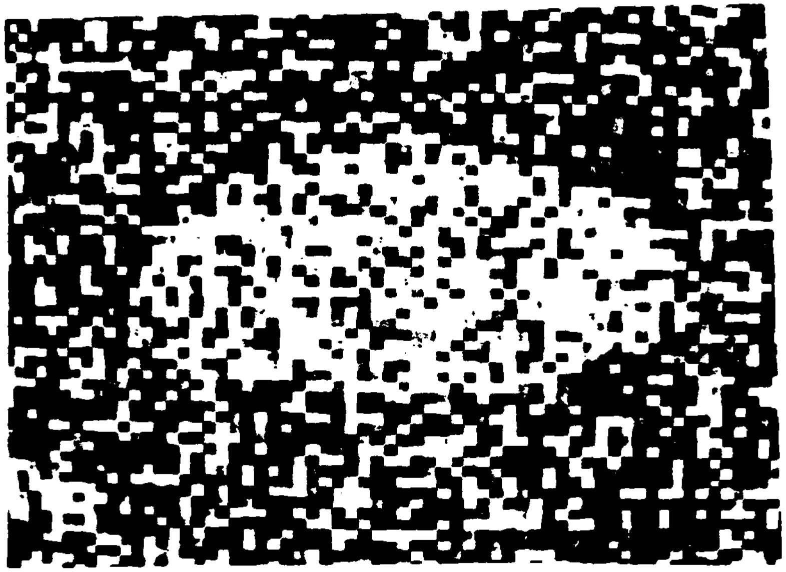


Figure 3a. 64x64 ellipse in additive Gaussian noise such that $\Delta/\sigma = 1$.

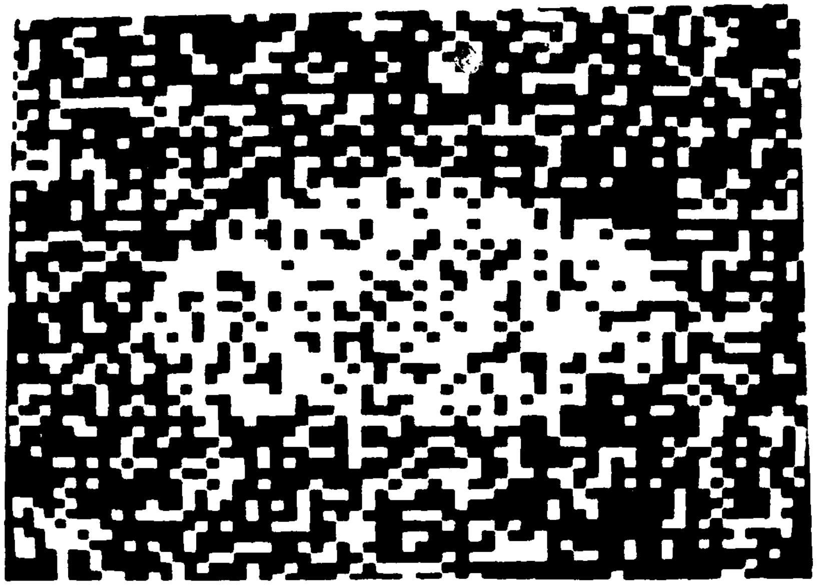


Figure 3b. Maximum likelihood threshold segmentation.

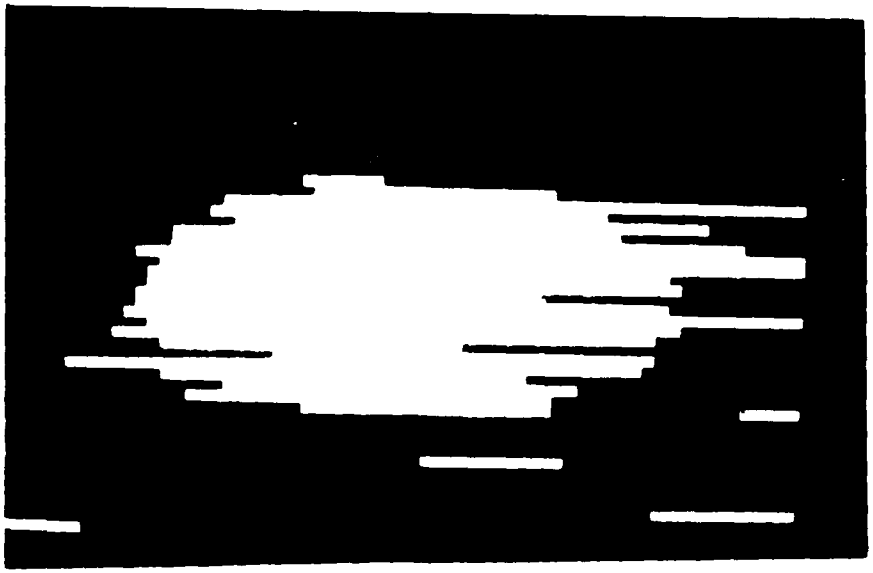


Figure 3c. MAP segmentation with T=.95 and M=1.

Figure 3d. MAP segmentation with T=.95 and M=4.

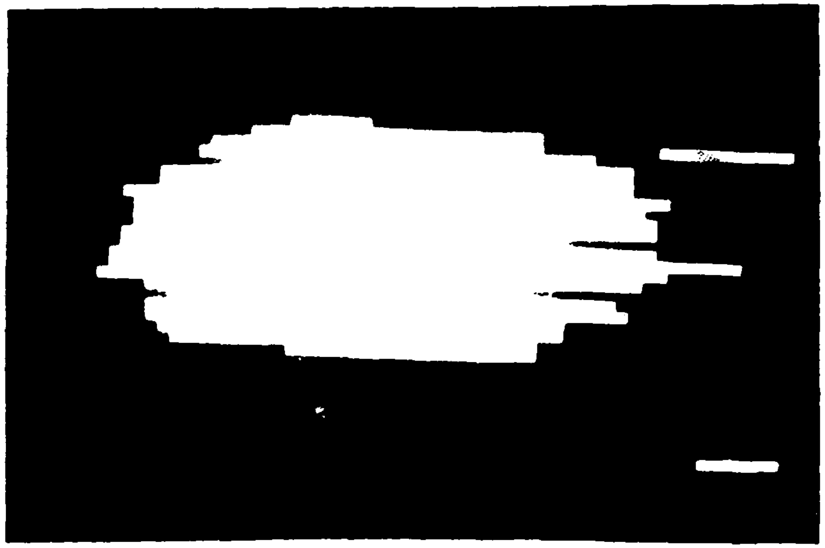


Figure 3e. MAP segmentation with T=.95 and M=8.

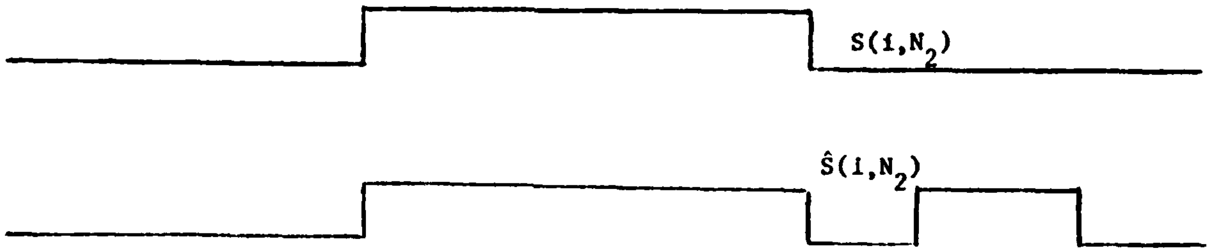


Figure 4a. Intensity profiles for correct row segmentation $S(i,N_2)$ and incorrect segmentation $\hat{S}(i,N_2)$ for case 1.

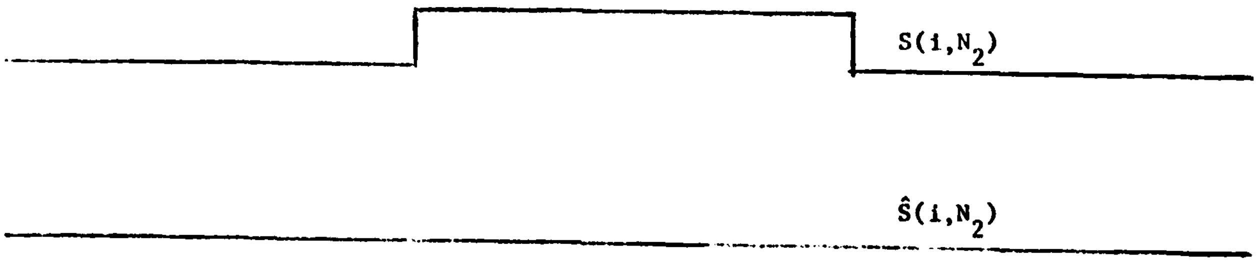


Figure 4b. Intensity profiels for correct row segmentation $S(i,N_2)$ and incorrect segmentation $\hat{S}(i,N_2)$ for case 2.

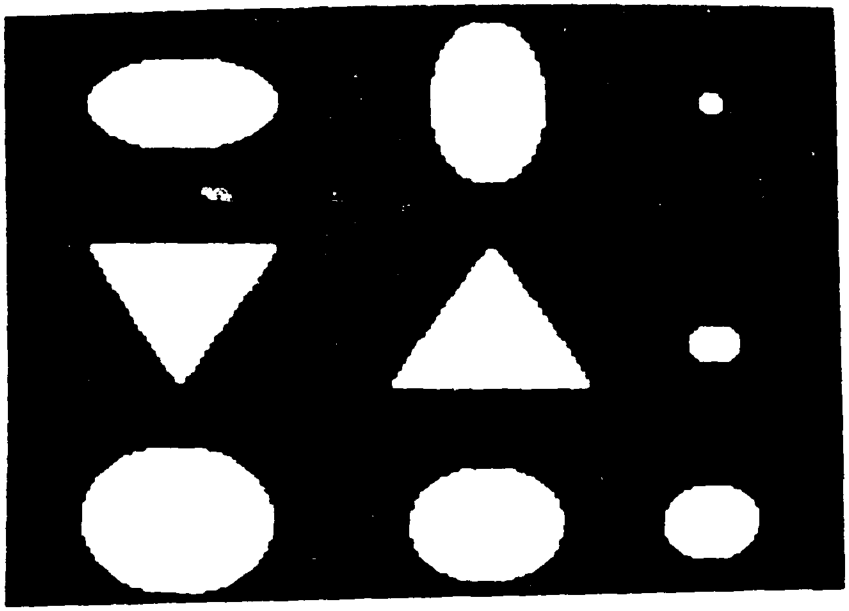


Figure 5a. 170x170 picture containing two ellipses, two triangles and four circles with radii ranging from 2 pixels to 20 pixels.

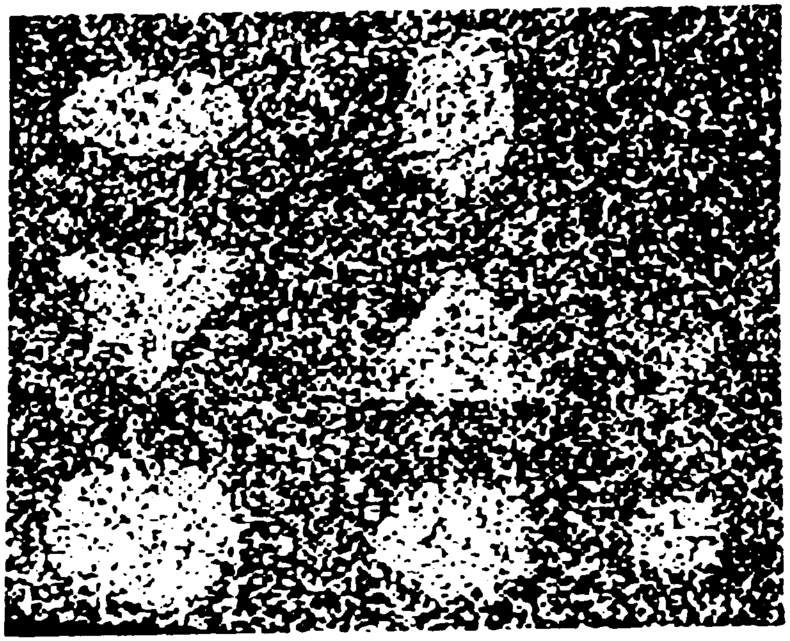


Figure 5b. Picture 5b with additive Gaussian noise such that $\Delta/\sigma = 1$.

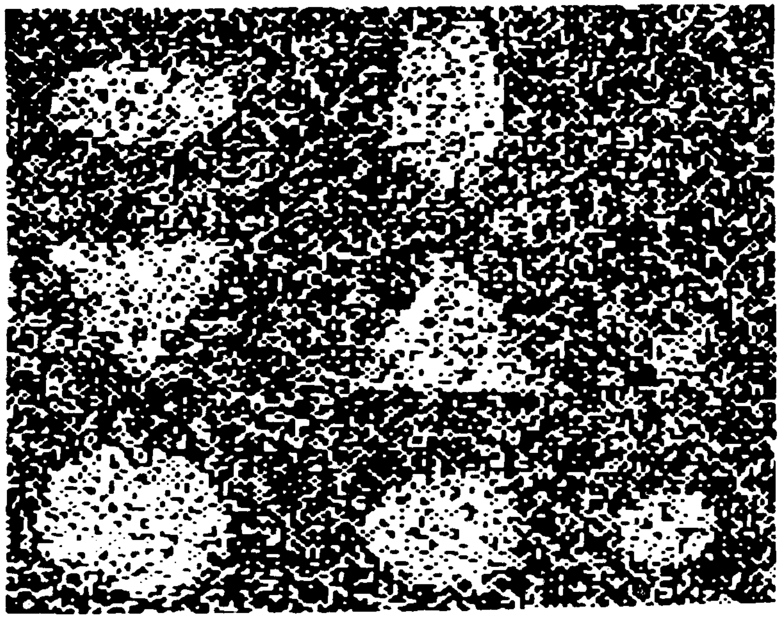


Figure 5c. MAP segmentation with T=.5
(equivalent to thresholding).

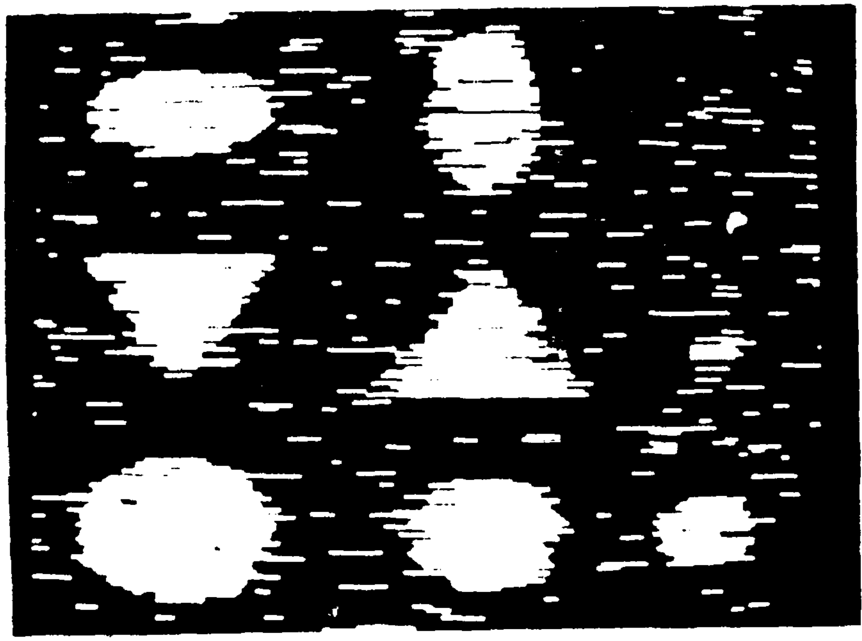


Figure 5d. MAP segmentation with T=.8 and M=8.

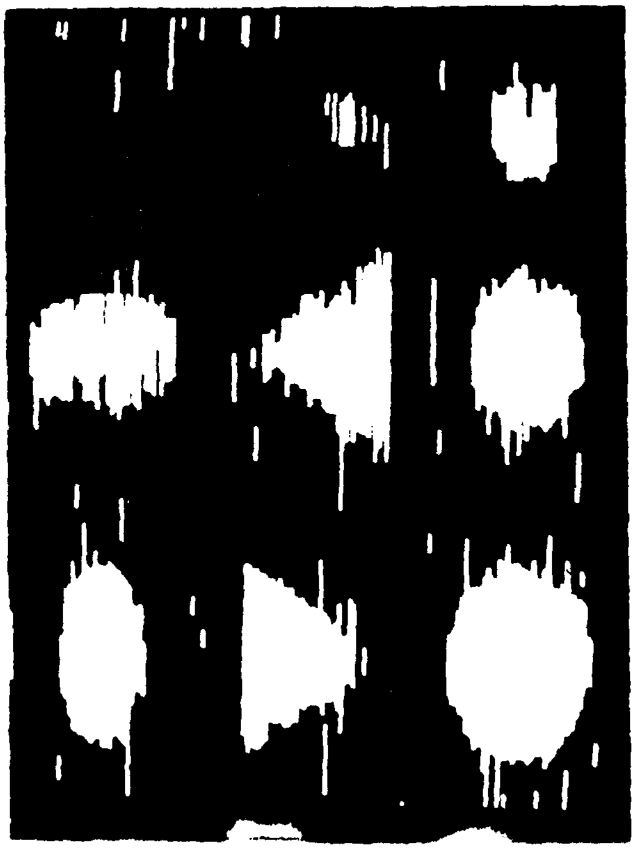


Figure 5e. MAP segmentation with T=.9 and M=8.

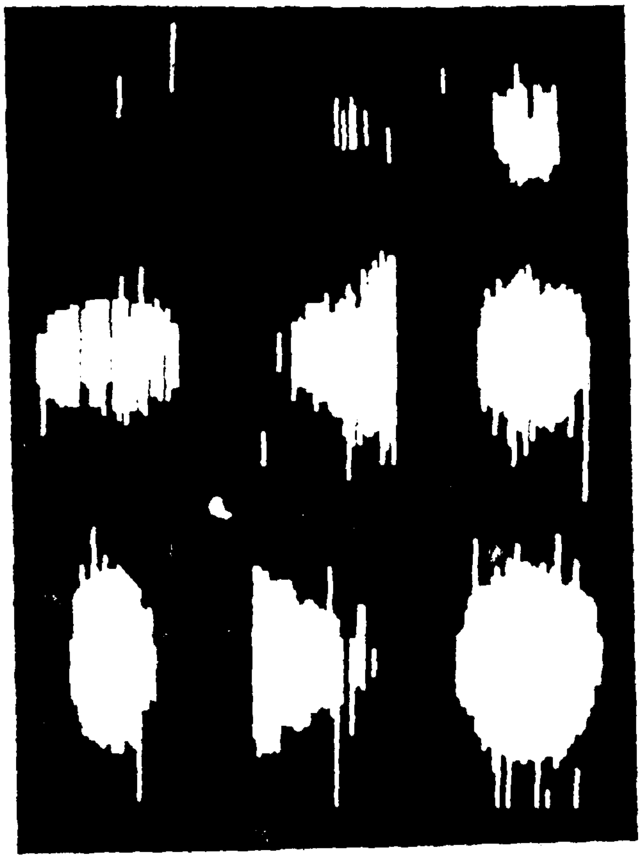


Figure 5f. MAP segmentation with T=.95 and M=8.

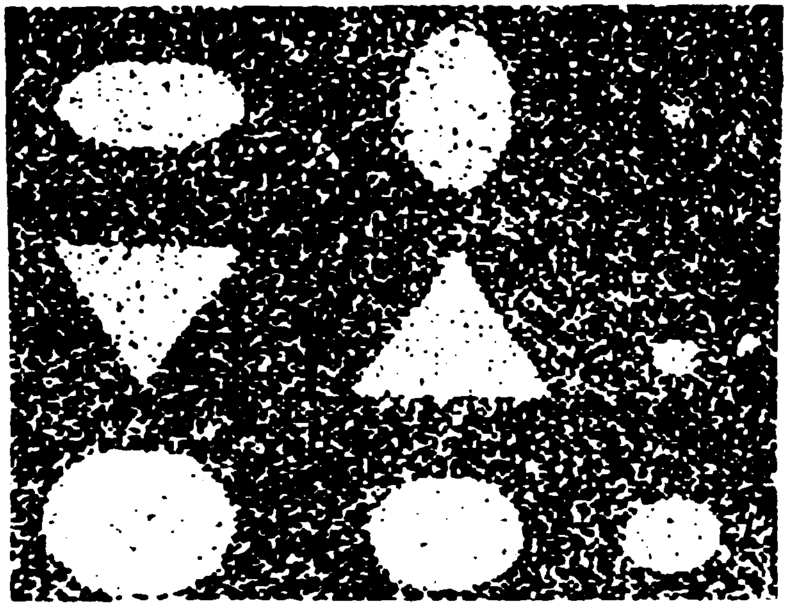


Figure 6a. Picture 5a with additive Gaussian noise such that Δ/σ = 2.

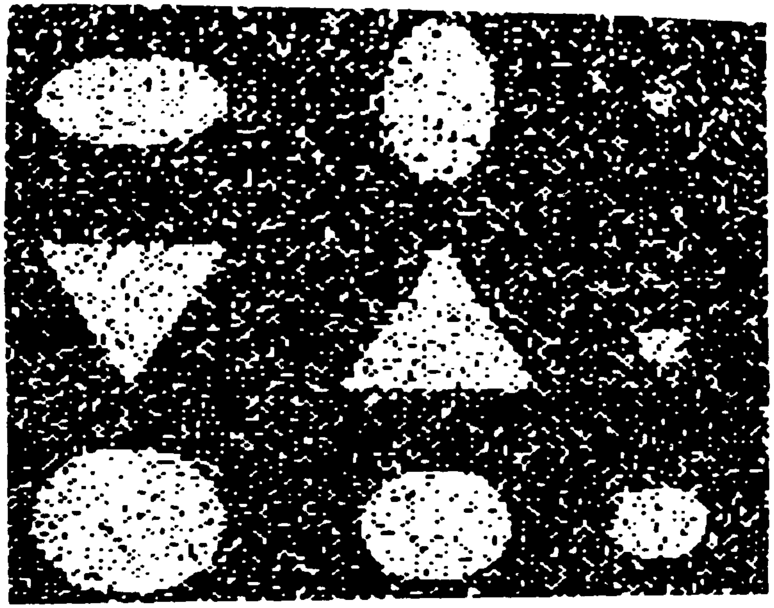


Figure 6b. MAP segmentation with T=.5 (equivalent to thresholding).

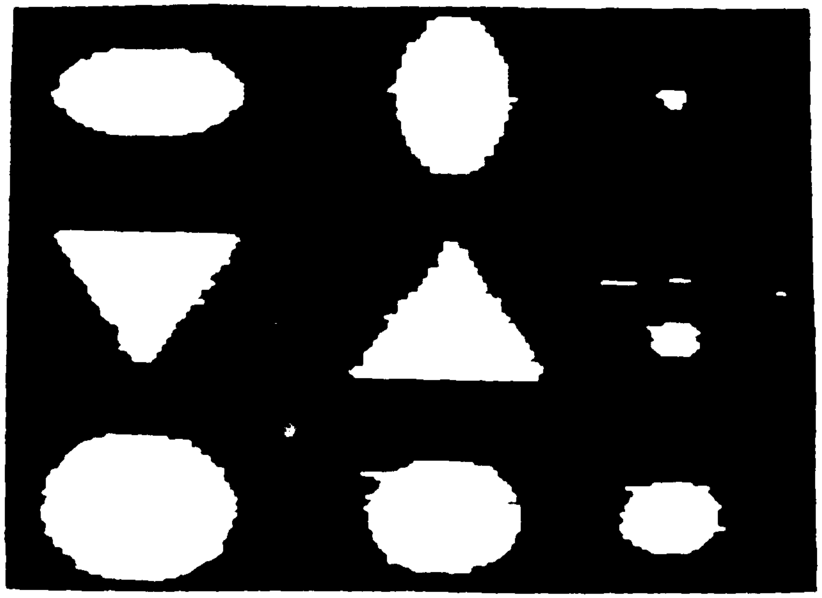


Figure 6c. MAP segmentation with T=.95 and M=8.

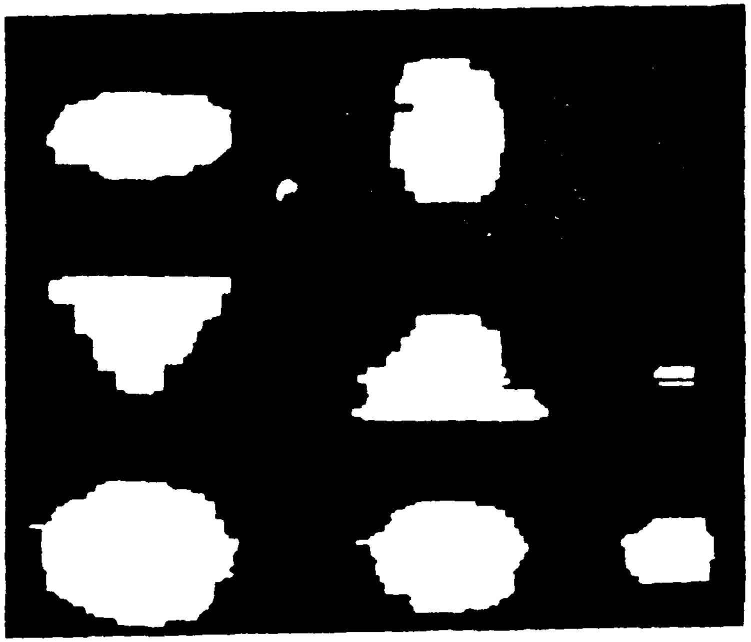


Figure 7a. Post-filtered version of 5f.

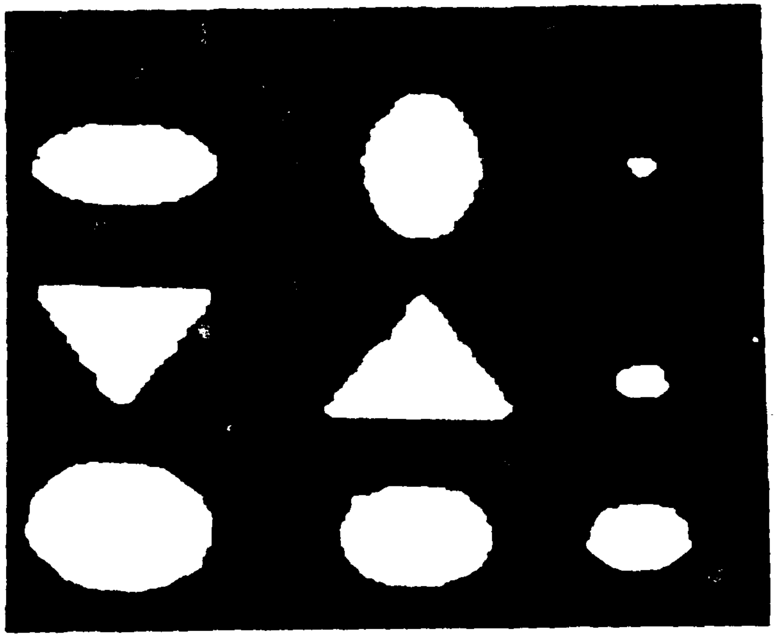


Figure 7b. Post-filtered version of 6c.

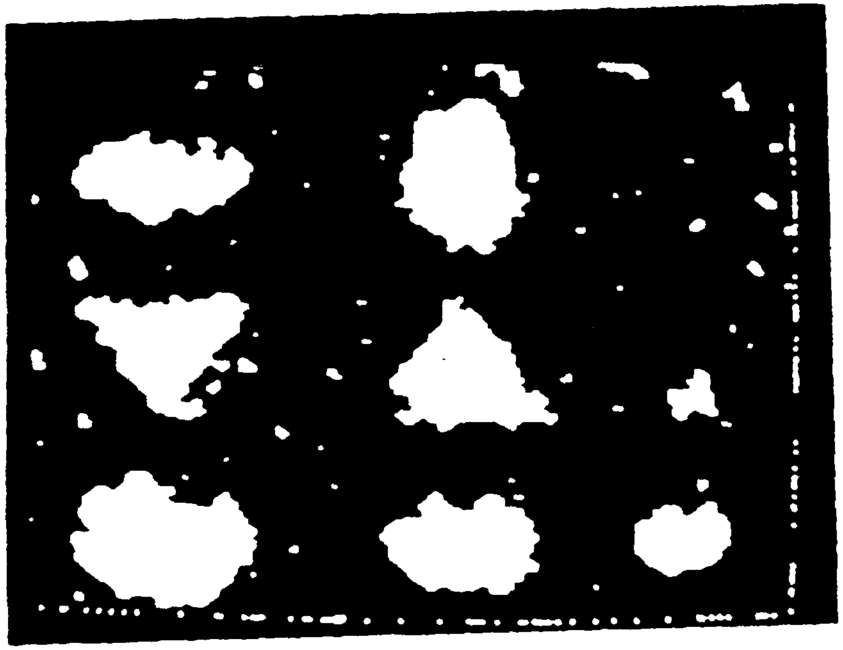


Figure 7c. Post-filtered version of 5c.

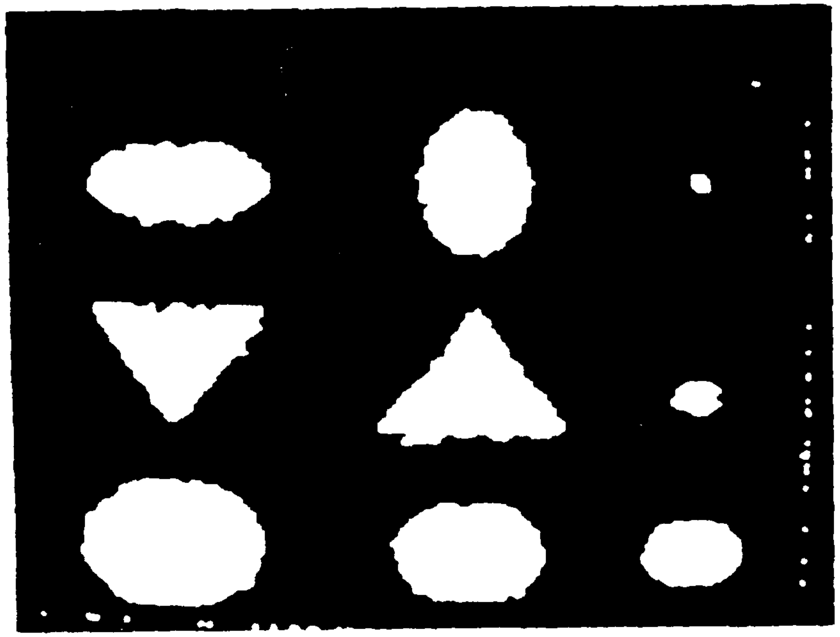


Figure 7d. Post-filtered version of 6b.

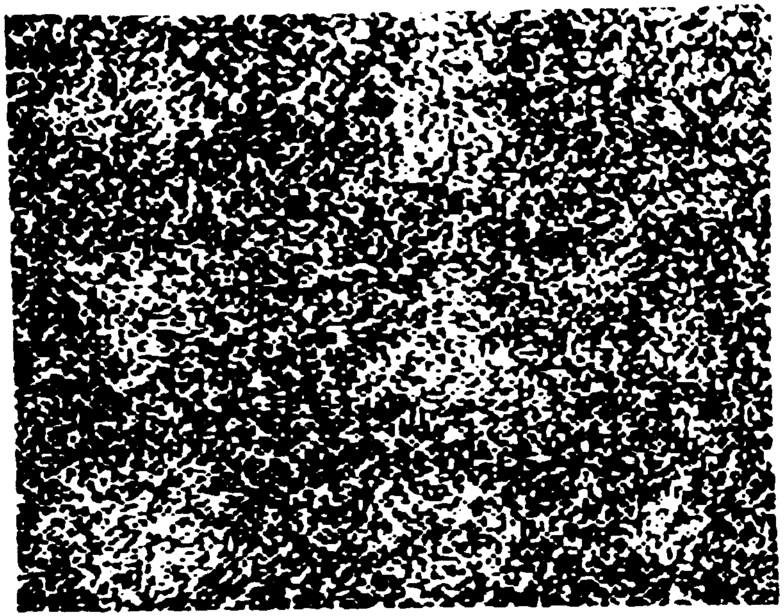


Figure 8a. Picture 5a with additive Gaussian noise such that Δ/σ = .8.

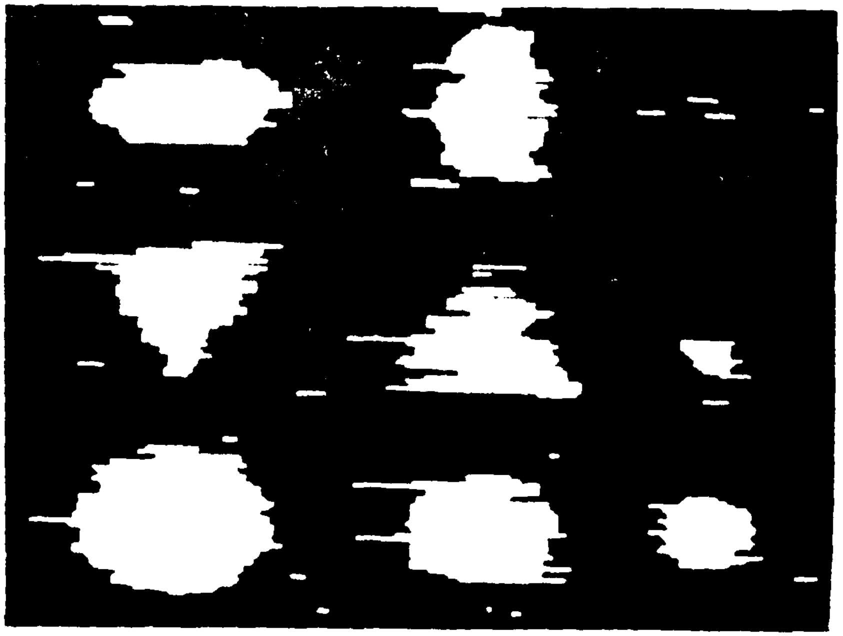


Figure 8b. Modified MAP segmentation with T=.83, T_R=.55 and M=8.

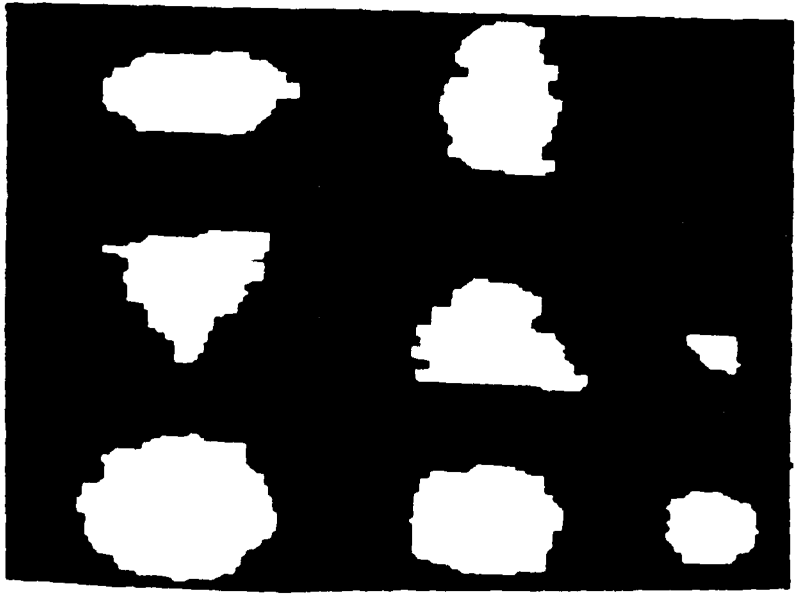


Figure 8c. Post-filtered version of 8b.

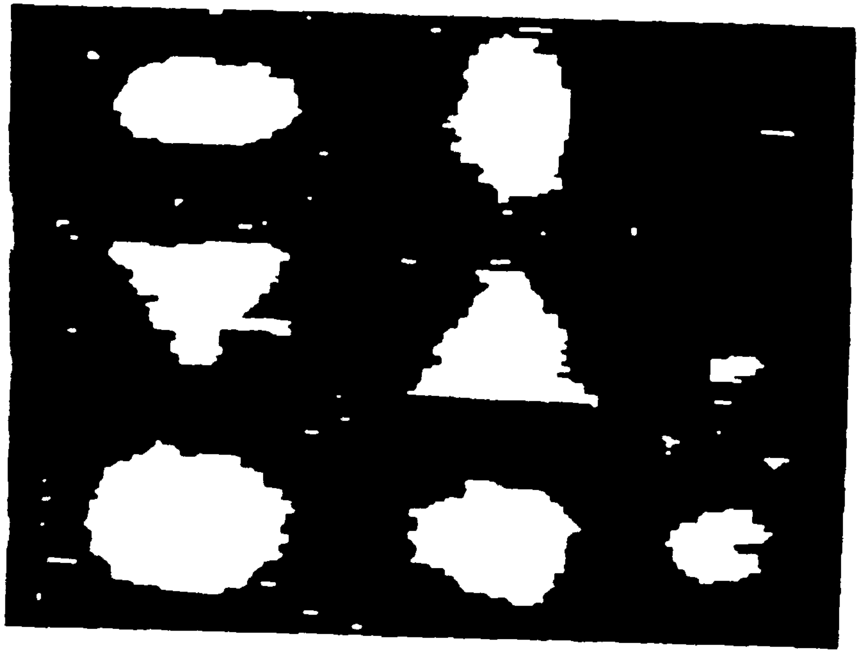


Figure 9a. Modified MAP segmentation of 5b with $T=.83$, $T_R=.6$ and $M=8$.

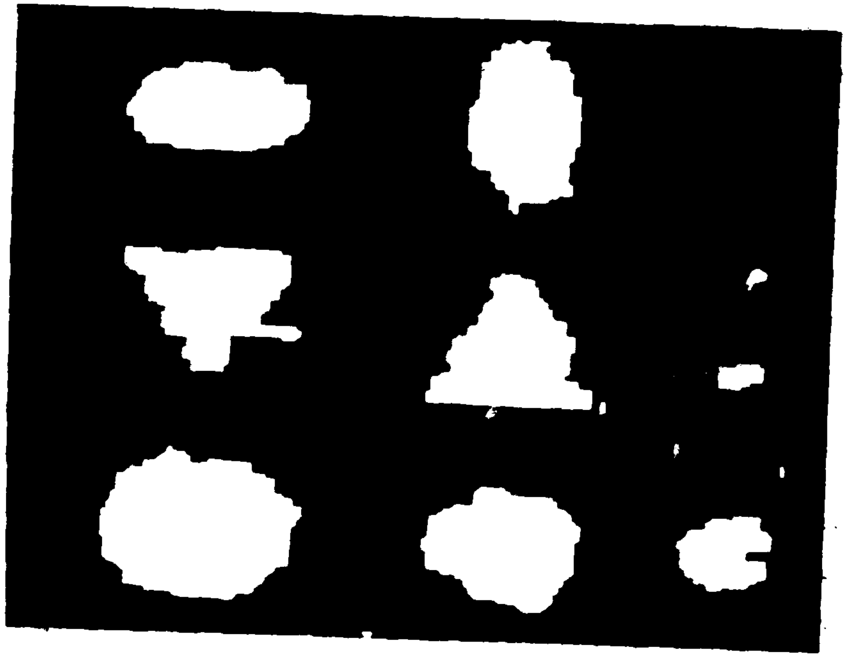


Figure 9b. Post-filtered version cf 9a.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER JU81-DELENG-1R	2. GOVT ACCESSION NO. AD-A112	3. RECIPIENT'S CATALOG NUMBER 914
4. TITLE (and Subtitle) Image Segmentation Using Simple Markov Field Models		5. TYPE OF REPORT & PERIOD COVERED Final
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) F. R. Hansen and H. Elliott		8. CONTRACT OR GRANT NUMBER(s) N00014-82-K-0076
9. PERFORMING ORGANIZATION NAME AND ADDRESS Department of Electrical Engineering Colorado State University Fort Collins, CO 80523		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Statistics and Probability Program Office of Naval Research, Code 436 Arlington, VA 22217		12. REPORT DATE Oct. 1981
		13. NUMBER OF PAGES 40
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release: Distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Markov fields, dynamic programming, image segmentation

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

By modelling a picture as a two-state Markov field, MAP estimation techniques are used to develop suboptimal but computationally tractable binary segmentation algorithms. The algorithms are shown to perform well at low signal to noise ratios, and analytical procedures are developed for estimating the Markov field transition probabilities. In addition, extensions of this approach to the multi-spectral and multi-region cases are discussed.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

DAT

FILM

4