AN ADAPTIVE CLUSTERING ALGORITHM FOR IMAGE SEGMENTATION

Thrasyvoulos N. Pappas and N. S. Jayant

Signal Processing Research Department
AT&T Bell Laboratories
Murray Hill, New Jersey 07974

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 8-neighbor Gibbs random field model applied to pictures of industrial objects and a variety of other images show that the algorithm performs better than the K-means algorithm and its nonadaptive extensions.

1 Introduction

In this paper we develop an algorithm for segmenting images of objects with smooth surfaces. The intensity of each region is assumed to be a slowly varying function plus noise, but no texture. Our goal is to separate the pixels in the image into clusters based on both their intensity and their relative location. To make use of the spatial information we assume 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]. 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 [2], Besag [3], Derin and Elliott [4], 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 a probability density function which has two components. One constrains the region intensity to be close to the data. The other imposes spatial continuity. The algorithm alternates between maximizing the 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 progressively decreases. Thus the algorithm starts with global estimates and slowly adapts to the local characteristics of each region.

The performance of this technique is clearly superior to K-means clustering for the class of images we are considering. It is also better than the nonadaptive extensions of K-means clustering that incorporate spatial constraints by the use of Gibbs random fields. 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. Our experimental results indicate that the performance of the algorithm is also superior to existing region growing techniques (see [5] for a survey; other thresholding techniques for image segmentation can be found in [6]). Applications of this technique include segmentation of pictures of industrial objects and adaptive thresholding of images.

The model for the class of images we are considering is presented in section 2. The algorithm is presented in section 3. In section 4 we show some examples. The conclusions of the paper are summarized in section 5.

2 Model

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. In this section we develop a model for the a *posteriori* probability density function $p(x|y)$. By Bayes' theorem

$$p(x|y) \propto p(y|x)\,p(x) \tag{1}$$

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

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

We consider images defined on the Cartesian grid and a neighborhood consisting of the 8 nearest pixels. According to the Hammersley-Clifford theorem [7] the density of x is given by a Gibbs density which has the following form

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

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 two-point cliques. They are defined as follows

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

The parameter β is positive, so that two neighboring pixels are more likely to belong to the same class than to different classes. Increasing the value of β has the effect of increasing the size of the regions and smoothing their boundaries.

The conditional density is modeled as a white Gaussian process, with mean μ_s^i and variance σ^2. Each region i is characterized by a different μ_s^i which is a slowly varying function of s. Thus the intensity of each region is modeled as a signal μ_s^i plus white

CH2673-2/89/0000-1667 $1.00 © 1989 IEEE

Gaussian noise with variance σ^2.

The combined probability density has the form

$$p(x|y) \propto \exp\left\{-\sum_s \frac{1}{2\sigma^2}[y_s - \mu_s^{x_s}]^2 - \sum_C V_C(x)\right\} \quad (5)$$

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 μ_s^i do not depend on s, then we get a model similar to those used in [2, 3, 4]. Moreover, if $\beta = 0$, 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.

For the time being we assume that the parameter β and the noise variance σ^2 are known. Thus we must estimate both the distribution of regions x and the intensity functions μ_s^i. This will be done in the following section.

3 Algorithm

In this section we consider an algorithm for estimating the distribution of regions x and the intensity functions μ_s^i. Note that the functions μ_s^i are defined on the same grid as the original grey level 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.

The initial estimate of x is given by the K-means algorithm. Given the region labels x, we estimate the intensity μ_s^i at each pixel s as follows. We average the grey levels of all the pixels that belong to region i and are inside a window of width W centered at pixel s. We have to obtain estimates of μ_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 μ_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). Note that the functions μ_s^i are smooth and therefore the interpolated values are good approximations of the values we would obtain by an exact calculation. Also the functions are smoother for larger windows, hence the dependence of the spacing on the window size.

Given the intensity functions μ_s^i, we want to maximize (5) 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 [2]. Instead, we maximize the conditional density at each point x_s given the data y and the current x at all other points

$$p(x_s|y, x_q, \text{all } q \neq s) = p(x_s|y_s, x_q, q \in N_s)$$
$$\propto \exp\left\{-\frac{1}{2\sigma^2}[y_s - \mu_s^{x_s}]^2 - \sum_{x_s \in C} V_C(x)\right\} \quad (6)$$

The equality on the left follows from the Markovian property and the whiteness of the noise. The maximization is done at all points in the image and the cycle is repeated until convergence. This is the Iterated Conditional Modes (ICM) approach proposed by Besag [3], sometimes also called the greedy algorithm [8]. It corresponds to instantaneous freezing in simulated annealing.

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 reliable and more 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 Figure 1.

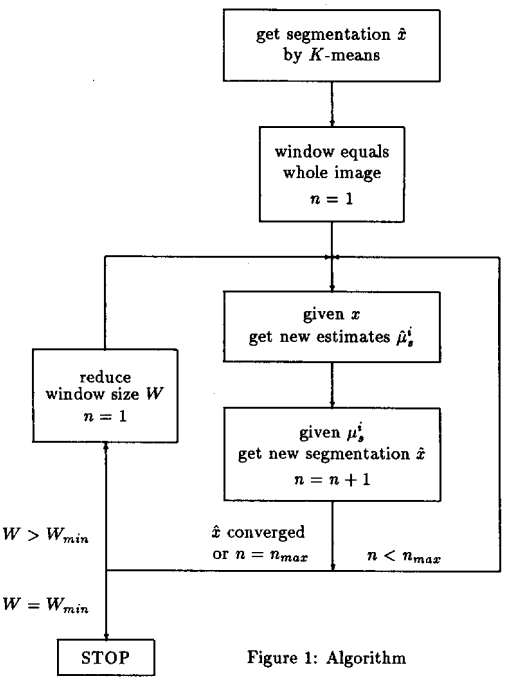

Figure 1: Algorithm

As the size of the window for estimating the intensity functions decreases, it is possible that there are not enough pixels of type i inside the window centered at s to estimate μ_s^i. In that case μ_s^i is undefined and x_s cannot be assigned to i in the next iteration. The minimum number of pixels that are necessary for estimating μ_s^i was chosen equal to W. The higher the threshold the more robust the computation of the μ's. On the other hand, when the threshold is high, some small, isolated regions may disappear.

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 [5, 6], 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.

We assume that the noise variance σ^2 can be estimated independently of the algorithm. From (5) it follows that increasing σ^2 is equivalent to decreasing β. In fact, the parameter β 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) [3, 9].

The amount of computation for the estimation of x depends on the number of cycles. An iteration converges when the number of pixels that change during a cycle is less then a threshold (typically $M/10$ for an $M \times M$ image). This usually requires a few cycles. For a given window, the number of iterations required is typically less than 10. The convergence criterion is that the last iteration converges in one cycle. The amount of computation for each local intensity calculation does not depend on the window size because the grid spacing depends on the window size. Typically we keep reducing the window size by a factor of 2 or

$\sqrt{2}$, until a minimum window size of 7 pixels.

Finally we should point out that the 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.

4 Examples

An original image is shown in Figure 2(a); the resolution is 256×256 pixels and the grey level range is 0 to 255 (8 bits). Figure 2(b) shows the result of the K-means algorithm. Figure 2(c) shows the result of our adaptive algorithm. In both cases we chose $K = 4$. We can clearly see the superior performance of the adaptive algorithm.

Another original image is shown in Figure 3(a); the resolution is 256×256 pixels. White Gaussian noise with $\sigma = 7$ grey levels has been added to this image. Figure 3(b) shows the result of the K-means algorithm. Figure 3(c) shows the result of our adaptive algorithm. Again we chose $K = 4$. Here, too, the superior performance of the adaptive algorithm is obvious.

In the examples above we set the number of classes $K = 4$. In many clustering algorithms the choice of the number of classes can be crucial. In the case of the K-means algorithm, choosing the wrong number of classes can be disastrous as we can see in Figure 3(d), where $K = 2$. Because of the wide variation of the grey level throughout the image, a lot of detail has been lost in the K-means case. 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. This is illustrated in Figure 3(e), where most of the detail in the image has been recovered.

5 Conclusion

We presented an algorithm for segmenting images of objects with smooth surfaces. The intensity of each region was modeled as a slow varying function plus white Gaussian noise, and the underlying region distribution was modeled by a Gibbs random field. The adaptive technique we developed is a generalization of the K-means algorithm. We applied our algorithm to a variety of pictures with considerable success. We believe that our approach can be extended to the case of images that include regions with slowly varying textures.

References

[1] J. T. Tou and R. C. Gonzalez, *Pattern Recognition Principles*, Addison, 1974.

[2] S. Geman and D. Geman, "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images," *IEEE Trans. Pat. Anal. Mach. Int.*, Vol. PAMI-6, No. 6, pp. 721–741, November 1984.

[3] Julian Besag, "On the Statistical Analysis of Dirty Pictures," *J. Royal Statist. Soc. B*, **48**, *No. 3*, pp. 259–302, 1986.

[4] H. Derin and H. Elliott, "Modeling and Segmentation of Noisy and Textured Images Using Gibbs Random Fields," *IEEE Trans. Pat. Anal. Mach. Int.*, Vol. PAMI-9, No. 1, pp. 39–55, January 1987.

[5] R. M. Haralick and L. G. Shapiro, "Image Segmentation Techniques," *Computer Vision, Graphics, and Image Processing*, Vol. 29, pp. 100–132, 1985.

[6] P. K. Sahoo, S. Soltani, A. K. C. Wong, "A Survey of Thresholding Techniques," *Computer Vision, Graphics, and Image Processing*, Vol. 41, pp. 233–260, 1988.

[7] Julian Besag, "Spatial Interaction and the Statistical Analysis of Lattice Systems," *J. Royal Statist. Soc. B*, **26**, *No. 2*, pp. 192–236, 1974.

[8] H. Derin, "Experimentations with Simulated Annealing in Allocation Optimization Problems," *Proceedings CISS-86*, pp. 165–171.

[9] G. Wolberg and T. Pavlidis, "Restoration of Binary Images Using Stochastic Relaxation with Annealing," *Pattern Recognition Letters*, Vol. 3, No. 6, pp. 375–387, December 1985.

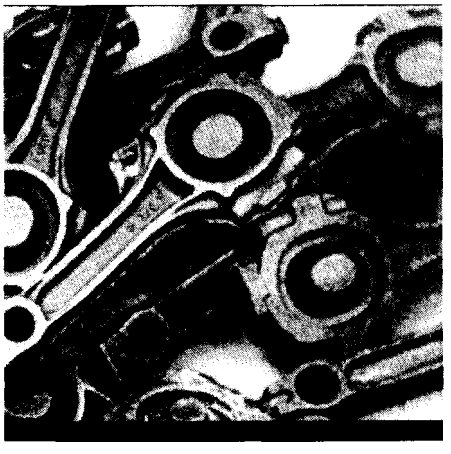

Figure 2(a): Original

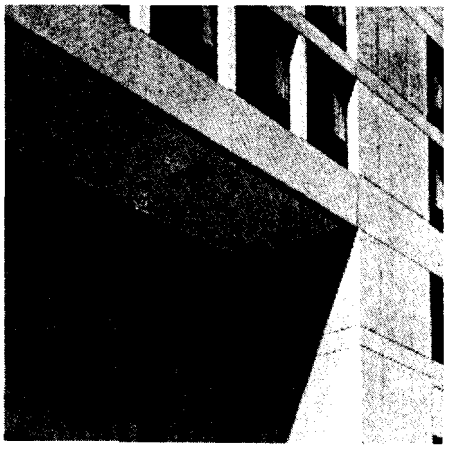

Figure 3(a): Original

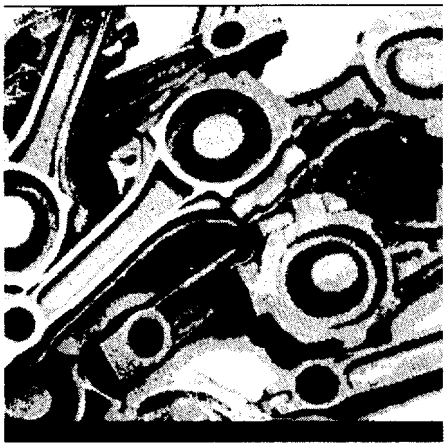

Figure 2(b): K-Means Algorithm ($K = 4$ Levels)

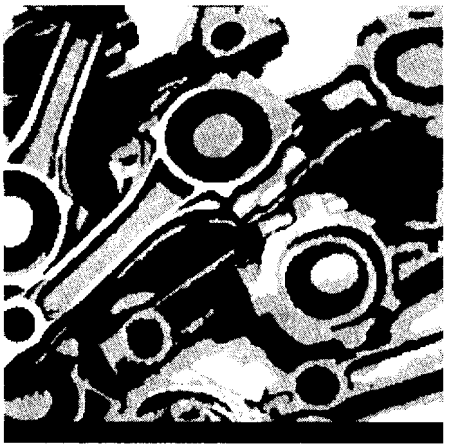

Figure 2(c): Adaptive Clustering ($K = 4$ Levels)

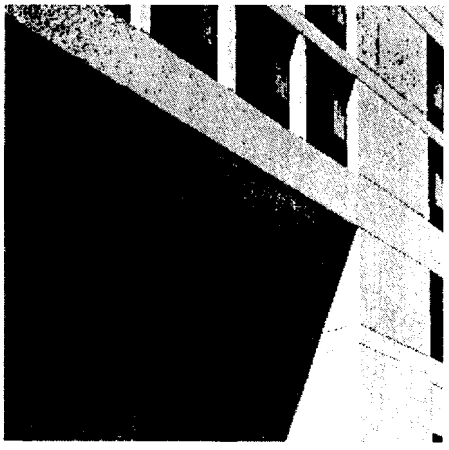

Figure 3(b): K-Means Algorithm ($K = 4$ Levels)

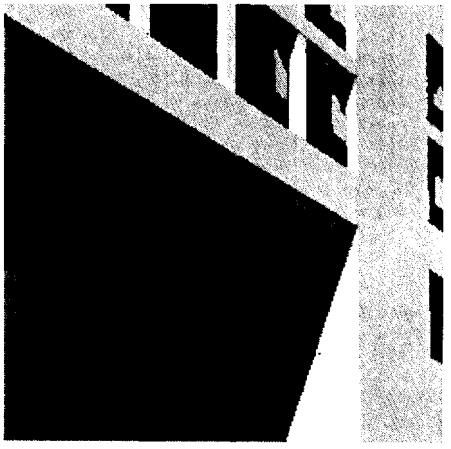

Figure 3(c): Adaptive Clustering ($K = 4$ Levels)

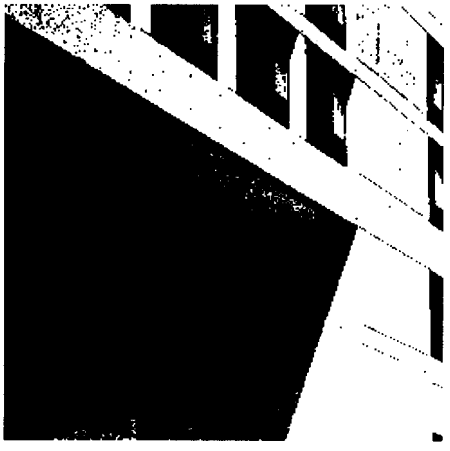

Figure 3(d): K-Means Algorithm ($K = 2$ Levels)

Figure 3(e): Adaptive Clustering ($K = 2$ Levels)