Proc. ICASSP-95, vol. 4, (Detroit, MI), pp. 2427–2430, May 1995.

AN ADAPTIVE CLUSTERING ALGORITHM FOR SEGMENTATION OF VIDEO SEQUENCES

Raynard O. Hinds

EECS Department
MIT, Cambridge, MA 02139

Thrasyvoulos N. Pappas

Signal Processing Research Department
AT&T Bell Labs, Murray Hill, NJ 07974

ABSTRACT

We present a Bayesian approach for segmenting a sequence of gray-scale images to obtain a binary sketch. We extend a 2-D algorithm to video sequences. The 2-D algorithm is an adaptive thresholding scheme that uses spatial constraints and takes into consideration the local intensity characteristics of the image. We model the segmentation distribution as a 3-D Gibbs Random Field. We add temporal constraints and temporal local intensity adaptation to ensure a smooth transition of the segmentation from frame to frame. For computational efficiency as well as performance we use a multi-resolution approach. We also consider several suboptimal implementations to reduce the delay as well as the amount of computation. We tested the performance of the algorithm on head and shoulders video sequences. The algorithm achieves accurate rendering of the lip and eye movements and preserves the main characteristics of the face, so that it is easily recognizable.

1. INTRODUCTION

We consider the problem of obtaining a binary sketch of a sequence of gray-scale images. We use a Bayesian approach to segment the gray-scale images into black and white regions. Each segmented image preserves significant features while discarding unimportant detail. We extend the 2-D adaptive clustering algorithm of [1] to video sequences. The 2-D (still image) algorithm is an adaptive thresholding scheme that uses spatial constraints and takes into consideration the local intensity characteristics of the image. We model the segmentation distribution as a 3-D Gibbs Random Field. We add temporal constraints and temporal local intensity adaptation to ensure a smooth transition of the segmentation from frame to frame. Similar approaches have been considered for segmentation of 3-D medical image data in [2] and [3]. However, there are significant differences between video sequences and 3-D still images.

For computational efficiency as well as performance we use a multi-resolution approach. We also consider several suboptimal implementations to reduce the delay as well as the amount of computation.

We tested the algorithm on several video sequences and showed that the sketches it creates move smoothly in time without limiting the temporal resolution. In contrast, when the 2-D algorithm [1] is applied to each frame independently, it results in unpleasant temporal artifacts. For head and shoulders sequences, the algorithm achieves accurate rendering of lip and eye movements and preserves the main characteristics of the face, so that it is easily recognizable.

The moving sketches can be displayed using a simple binary display device with limited spatial resolution. Also, since the raw data rate for each frame is low, the moving sketches make it possible to achieve high frame rates when the overall data transfer rate is limited. Finally, the sketches may be compressed efficiently using a number of techniques [4].

2. MODEL

A spatio-temporal 3-D volume is constructed from a sequence of gray-scale images, where each image is a 2-D slice in the volume. We model this stack of gray-scale images as a collection of regions of uniform or slowly varying intensity. Each region varies in shape and size and extends throughout the sequence of images. The only sharp transitions in gray level occur at the region boundaries.

Let $\mathbf{y}$ be the 3-D volume of images and y_t be the observed gray-scale image at time t. Each 2-D slice consists of a grid of sites s and the intensity of a pixel at site s is denoted by $y_{s,t}$. The pair (s,t) can index each location in the 3-D volume. A segmentation of the image sequence into regions will be denoted by $\mathbf{x}$, where x_t is the segmentation of the image y_t into regions. Let $x_{s,t} = i$ mean that the pixel at site s and time t belongs to region i. The number of different region types is K. We set $K = 2$ to obtain binary images.

We develop a model for the *a posteriori* probability density function $p(\mathbf{x}|\mathbf{y})$. By Bayes' theorem

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

where $p(\mathbf{x})$ is the *a priori* density of the region process and $p(\mathbf{y}|\mathbf{x})$ is the conditional density of the observed sequence of images given the distribution of regions. We model the region process by a 3-D Gibbs random field (GRF). It satisfies the Markovian property, that is, if $N_{s,t}$ is a neighborhood of the pixel at site s and time t, then

$$p(x_{s,t}|x_{q,r}, \text{all } (q,r) \neq (s,t)) = p(x_{s,t}|x_{q,r}, (q,r) \in N_{s,t})$$

We consider each image defined on the Cartesian grid and a neighborhood consisting of the 8 nearest pixels in the same 2-D slice and the two adjacent pixels at the identical site s in the surrounding frames. The Gibbs density for $\mathbf{x}$ has the following form

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

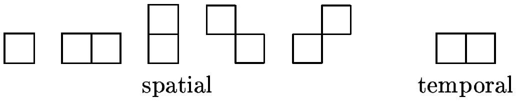

Figure 1: Clique types for Gibbs density

where Z is a normalizing constant, $V_C(\mathbf{x})$ are the clique potentials, and the summation is over all cliques C. A clique is a set of points that are neighbors of each other. A clique potential V_C is a function that depends only on the pixels that belong to a clique C.

Our model assumes that the only nonzero potentials are those that correspond to the one- and two-point cliques shown in Fig. 1. Notice that in this simple case the cliques are either spatial (S) or temporal (T). The two-point clique potentials are defined as follows:

$$V_S(\mathbf{x}) = \begin{cases} -\beta_1, & \text{if } x_{s,t} = x_{q,t} \text{ and } (s,t),(q,t) \in S \\ +\beta_1, & \text{if } x_{s,t} \neq x_{q,t} \text{ and } (s,t),(q,t) \in S \end{cases}$$

$$V_T(\mathbf{x}) = \begin{cases} -\beta_2, & \text{if } x_{s,t} = x_{s,r} \text{ and } (s,t),(s,r) \in T \\ +\beta_2, & \text{if } x_{s,t} \neq x_{s,r} \text{ and } (s,t),(s,r) \in T \end{cases}$$

The parameters β_1 and β_2 are positive, so that two neighboring pixels are more likely to belong to the same class than to different classes. The clique potentials are intended to control the interaction between pixels within a single frame as well as across frames. We arbitrarily chose the total weight of the interaction within each frame to be equal to the total weight of the interaction across frames. Thus, $2\beta_2 = 8\beta_1$. Increasing the value of β_1 has the effect of increasing the size of the regions in each frame and smoothing their boundaries. We further assume that the one-point clique potentials are zero, which means that all region types are equally likely.

The conditional density is modeled as a white Gaussian process, with mean $\mu^i_{s,t}$ and variance σ^2. Each region i is characterized by a different $\mu^i_{s,t}$ which is a slowly varying function of s and t.

The combined probability density has the form

$$p(\mathbf{x}|\mathbf{y}) \propto \exp\left\{-\sum_{t,s} \frac{1}{2\sigma^2}\left[y_{s,t} - \mu^{x_{s,t}}_{s,t}\right]^2 - \sum_C V_C(\mathbf{x})\right\} \quad (1)$$

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 and temporal continuity. Note also that increasing σ^2 is equivalent to increasing β_1 and β_2. Thus, we fix β_1 and β_2 and estimate the noise variance σ^2. In [1] it is shown 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 in the segmentation.

3. ALGORITHM

We now consider an iterative algorithm for estimating the distribution of regions $\mathbf{x}$ and the local intensity functions $\mu^i_{s,t}$ throughout the 3-D volume. At each frame t, the algorithm alternates between estimating x_t and the intensity functions $\mu^i_{s,t}$. Note that the functions $\mu^i_{s,t}$ are defined on the same 3-D grid as the original gray-scale sequence $\mathbf{y}$ and the distribution of regions $\mathbf{x}$.

First, we consider the problem of estimating the local intensity functions in a frame at time t (denoted by μ^i_t). Given the region labels in the frame x_t and the two surrounding frames x_{t-1} and x_{t+1}, we estimate the intensity $\mu^i_{s,t}$ at each pixel s in the frame by averaging the gray levels of all the pixels that belong to region i and are inside a window of width W centered at pixel s in three consecutive frames.

As we saw in [1], when the total number of pixels of type i inside the windows centered at s in the three frames is too small, the estimate of $\mu^i_{s,t}$ is not reliable and hence $\mu^i_{s,t}$ is undefined. In such a case, $x_{s,t}$ cannot be assigned to level i. The algorithm specifies the minimum number of pixels T_{min} that are necessary for estimating $\mu^i_{s,t}$. The higher this threshold, the more robust the computation of $\mu^i_{s,t}$. A reasonable choice for this parameter is $T_{min} = 3W$, the sum of the window widths in the three frames. This value of the threshold guarantees that long one-pixel wide regions will be preserved.

The estimates of $\mu^i_{s,t}$ must be obtained for all region types i and all pixels s in each frame. As in [1], to reduce computation, we obtain the estimates $\mu^i_{s,t}$ only on a grid of points in each frame, and use bilinear interpolation to obtain the remaining values. The spacing of the grid points in a frame is a function of the window size. We chose the spacing equal to half the window size in each spatial dimension (50% overlap). Since the functions $\mu^i_{s,t}$ are smooth, this is a good approximation. It also guarantees that the amount of computation is independent of window size.

Second, we consider the problem of estimating the distribution of regions. Given the intensity functions $\mu^i_{s,t}$, we must maximize the *a posteriori* probability density (1) to obtain the MAP estimate of $\mathbf{x}$. As in [1], we use the Iterated Conditional Modes (ICM) approach proposed by Besag [5] to obtain a local maximum. That is, we maximize the conditional density at each point $x_{s,t}$ given the data $\mathbf{y}$ and the current segmentation $\mathbf{x}$ at all other points.

$$\begin{aligned} & p(x_{s,t}|\mathbf{y}, x_{q,r}, \text{ all } (q,r) \neq (s,t)) \\ & \quad = \quad p(x_{s,t}|y_{s,t}, x_{q,r}, (q,r) \in N_{s,t}) \\ & \quad \propto \quad \exp\left\{-\frac{1}{2\sigma^2}\left[y_{s,t} - \mu^{x_{s,t}}_{s,t}\right]^2 - \sum_{x_{s,t} \in C} V_C(\mathbf{x})\right\} \end{aligned}$$

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 3-D volume and the *cycle* is repeated until convergence.

Now we consider the overall adaptive clustering algorithm. A direct extension of the algorithm proposed in [1] would obtain an initial estimate of $\mathbf{x}$ by the K-means algorithm. With this initial segmentation, the algorithm would alternate between estimating $\mathbf{x}$ and the intensity functions $\mu^i_{s,t}$. However, for computational efficiency as well as delay we consider several suboptimal algorithms.

Method A attempts to reduce the amount of computation while retaining the advantage of joint segmentation of all the frames in the 3-D volume. First, we obtain an initial estimate of $\mathbf{x}$ by the K-means algorithm applied to each frame individually. Given the region labels at some frame x_t and the two surrounding frames x_{t-1} and x_{t+1}, we estimate the intensity $\mu^i_{s,t}$ at each pixel s in the frame. Then we update the estimate of x_t using the ICM approach. The segmentation of the surrounding frames x_{t-1} and x_{t+1} is fixed. The algorithm then moves to the next frame and updates the estimates of μ^i_t and x_t, and so on, until all the frames

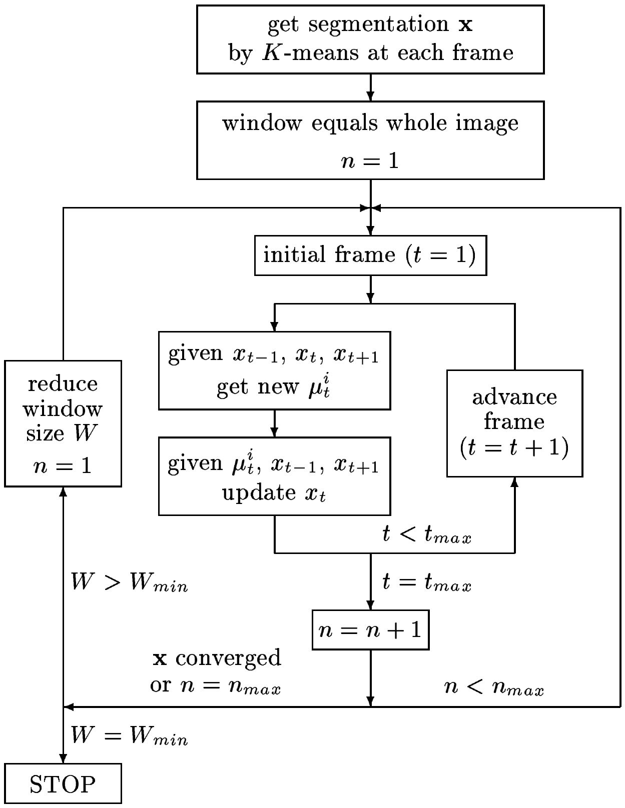

Figure 2: Adaptive clustering algorithm, Method A

are processed. Then the process is repeated. We define an *iteration* to consist of one update of $\mathbf{x}$ in all frames in the sequence. The window size for the intensity function estimation is kept constant until the procedure converges. Our stopping criterion is that the update of x_t at each frame in the volume converges in one cycle for all the frames. Weaker convergence criteria can be used to reduce the number of iterations. The whole procedure is then repeated with a smaller window size. The window depth (three consecutive frames) remains constant throughout the algorithm. The assumption is that scene characteristics remain fairly constant over time. A flowchart of the algorithm is given in Fig. 2. 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 size of $W = 7$ pixels.

In many applications, it is desirable to eliminate the delay that is necessary for the joint segmentation of all the frames in a sequence. *Method B* obtains the segmentation of each frame x_t successively, using only information from previous frames (i.e. it is causal). It is similar to the sequential method proposed in [2]. The GRF model is *approximated* by a truncated asymmetrical neighborhood since no information from future frames is available. The segmentation x_t for the starting image frame is determined using the 2-D algorithm [1]. The remaining frames are processed one at a time, using the K-means algorithm as an initial estimate of x_t. The algorithm alternates between estimating the local intensity functions μ_t^i and the segmentation x_t, using the segmentation of the previous frame x_{t-1} as a boundary condition. Only information from the current and previous frames is used. An *iteration* consists of one update of μ_t^i and x_t at a given frame. The window size for the intensity function estimation is kept constant until convergence. The whole procedure is repeated with a new window size until the minimum window size is reached. The final estimate of x_t is used as a boundary condition for the next frame.

Method C is a compromise between the first two methods. It segments the image frames successively, as in method B, while maintaining the joint estimation of x_t for consecutive frames, resulting in smoother motion, as in Method A. Method C estimates x_t using information from the previous, current, and next frame. It uses a symmetrical 3-D GRF model and a symmetrical window for the local intensity estimation. The segmentation x_t for the starting image frame is determined using the 2-D algorithm [1], as in Method B. For each of the remaining frames, x_t is obtained jointly with x_{t+1}, using x_{t-1} as a boundary condition. The initial estimates of x_t and x_{t+1} are obtained by the K-means algorithm. An *iteration* of the algorithm consists of an update of μ_t^i, followed by updates of x_t, then μ_{t+1}^i, and finally x_{t+1}. The window size for the intensity function estimation is kept constant until convergence. Then the whole procedure is repeated with a new window size until the minimum window size is reached. The final estimate of x_t is used as a boundary condition for the next frame and the estimate of x_{t+1} is disregarded.

Finally, as in [1], we use a *multi-resolution* approach to improve algorithm performance and computational efficiency. For each method, we construct a pyramid of images at different resolutions. For Method A, the algorithm as described above is performed on the images of the lowest resolution in the pyramid. When the minimum window size is reached, it moves to the next level in the pyramid and uses the current segmentation for all frames, expanded by two, as a starting point. As in [1], the starting window size for each level in the pyramid is twice the minimum window size of the previous level. For Methods B and C, the multiresolution implementation is applied when segmenting each frame.

4. EXPERIMENTAL RESULTS

We tested the algorithm on several video sequences. The spatial resolution is 180 × 120 pixels (QCIF) and the temporal resolution is 30 frames/second. The gray-scale resolution is 8 bits. We compared the performance of the different versions of our algorithm to a scheme that uses the adaptive clustering algorithm of [1] to segment each frame independently. We found that our algorithm creates binary sketches that move smoothly in time. In contrast, the independent scheme results in a lot of temporal artifacts (segments flashing on and off) that are unpleasant to watch. Some of these artifacts can be seen in Fig. 4 which shows two successive frames of the sequence processed by the independent scheme. Observe the discontinuity in the shadow under the chin and the sudden appearance of a white patch in the jacket. Such discontinuities are quite annoying when the sketches are displayed at 30 frames per second. The corresponding frames of the original gray-scale sequence are shown in Fig. 3. The results of the full 3-D adaptive clustering algorithm with symmetric temporal constraints (Method A) are shown in Fig. 5. Notice that the

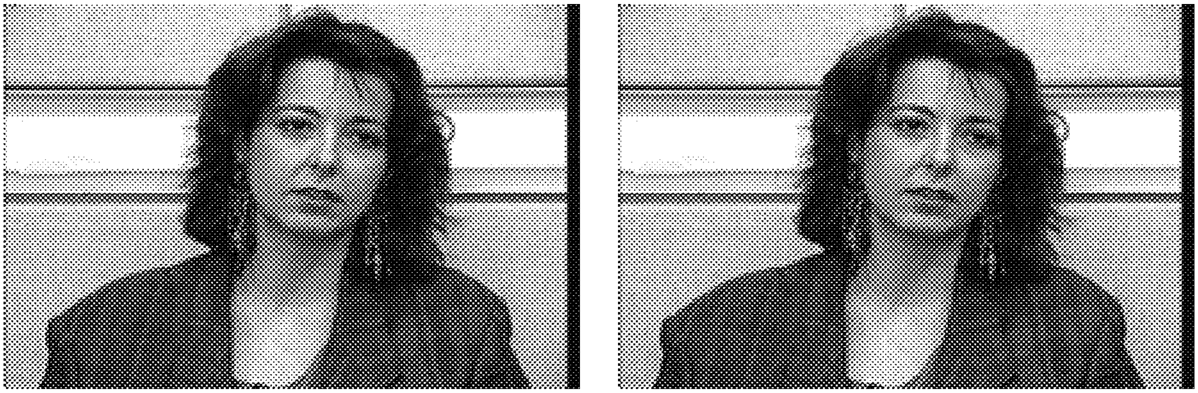

Figure 3: Gray-scale sequence

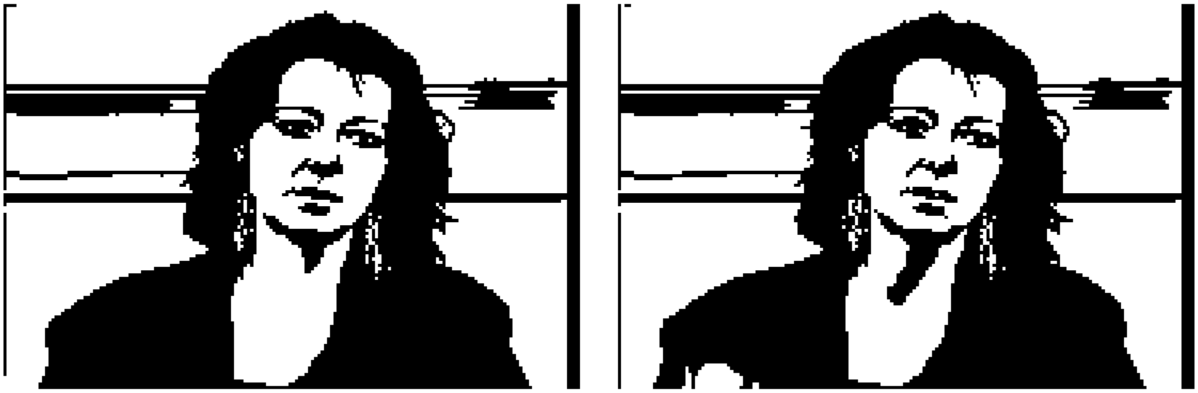

Figure 4: 2-D adaptive clustering algorithm (applied to each frame independently)

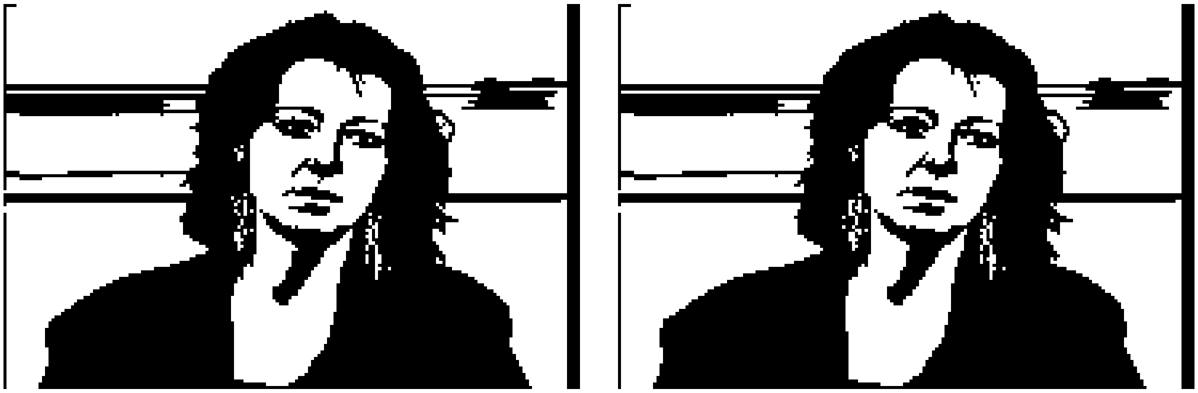

Figure 5: 3-D adaptive clustering algorithm with symmetrical temporal constraint (Method A)

discontinuities have now disappeared. We also found that the temporal constraints do not limit the temporal resolution of the moving sketches in any significant way. The lip and eye movements are very accurate. Only spurious transitions are eliminated.

The difference in performance between methods A and B is significant. Method A gives the best results but requires the whole sequence (or long segments thereof) at once, while method B requires only past frames. Method C requires only one look-ahead frame and its performance is very close to that of method A.

5. REFERENCES

[1] T.N. Pappas, "An adaptive clustering algorithm for image segmentation," *IEEE Tr. Signal Proc.*, vol. SP-40, p. 901–914, Apr. 1992.

[2] M.M. Chang, *et al.*, "Bayesian segmentation of MR images using 3-D Gibbsian priors," *Proc. SPIE, vol. 1903, Image and Video Proc.*, p. 122–133, 1993.

[3] C.W. Chen, *et al.*, "3D image segmentation via adaptive K-mean clustering and knowledge-based morphological operations," *IEEE Tr. Image Proc.* to appear.

[4] T.N. Pappas, "Adaptive thresholding and sketch-coding of grey level images," *Proc. SPIE, vol. 1199, Visual Comm. and Image Proc. IV*, p. 1003–1014, Nov. 1989.

[5] J. Besag, "On the statistical analysis of dirty pictures," *J. Royal Statist. Soc. B*, vol. 48, no. 3, p. 259–302, 1986.