SEGMENTATION OF AN IMAGE SEQUENCE USING MULTI-DIMENSIONAL IMAGE ATTRIBUTES (PROC. ICIP-96)

Edmond Chalom and V. Michael Bove, Jr.

MIT Media Laboratory,
20 Ames St., Cambridge, MA 02139, USA
{echalom,vmb}@media.mit.edu

ABSTRACT

Whether for purposes of compression efficiency, image editing, interactive multimedia authoring, or database search, it is often useful to be able to segment images or image sequences into regions corresponding to objects. In this paper we describe a segmentation scheme that takes account of multiple image characteristics, developing a multi-modal statistical model of regions based on a small amount of user-supplied training data.

1. INTRODUCTION

Researchers have shown the compression advantages of coding video as a set of regions that can be defined by motion or texture models. [1], [2], [3] Segmentation of video can likewise enable special effects (e.g. placing actors into a synthetic set) or authoring of interactive or personalizable programming. [4] In cases where semantics are as important as compression, though, one finds that generally objects in the real world do not correspond to a simple clustering model of a single parameter like motion; it is also true that the desired segmentation is a function of the application — for matting purposes it may be appropriate to regard a person as one region, while in a tele-shopping application with “hot buttons” a man’s shirt and trousers may need to be separate regions.

In previous work we have demonstrated segmentation based on a collection of image attributes. [5] This research also showed that supervised methods generally perform better than unsupervised algorithms, while simultaneously offering greater flexibility by permitting the user to define in advance the desired segmentation by indicating a few representative points corresponding to each region. Here we expand our method by generating more sophisticated statistical models for the regions.

2. FEATURE SELECTION

It is a natural, and almost unnoticed process for the human perceptual system adaptively to identify portions of a visual image as belonging to coherent objects in the world, based on a variety of static and dynamic image attributes. [6] Merely calculating some of these attributes remains a substantial task for machines, and even as increased computational power makes it easier, the greater task remains of intelligently assimilating the results (or “features”) in order to perform higher level image processing tasks. In this paper, the original raw image data is considered as a set of “observation” samples (or a vector of observations), and a “feature” vector is defined to be the result of a transformation or calculation applied to the observation vector. [7] The ideal feature space is one which simplifies the object segmentation process.

The raw image sequence observation vector is composed of RGB or (or YIQ, or YUV) pixel values, and it is necessary to perform transformations (which have physical interpretations associated with them) into the feature space to make the observation more useful. In a sense, there is more information embedded in the image sequence than meets the eye. In particular, it is possible to estimate the motion at every pixel directly from the observation data using a dense optical flow technique [8], [9] (indeed, a further transformation which would produce a higher-level motion model might be applied [2]). It is also possible to estimate texture information using local statistics on luminance values or other techniques such as SAR, [10] or steerable filters, [11] which depend only on the raw image data. Color and luminance information, as well as row and column positions at each pixel, can also be used as features, and require no additional calculation since it is obtained directly from the image data. Thus the physical attributes used in these experiments are motion, texture, color, and position.

The final assignment of a pixel into one or another
region is not completely dependent on the particular algorithm or a precise calculation of each feature. Specifically, because a combination of features is used, in conjunction with an intelligent modeling of the feature-space probability density function, the overall segmentation is quite robust and varies only slightly if a particular feature calculation is modified. For example, whether a block matching or an optical flow technique is used as the motion attribute, the segmentation result remains quite stable, provided the other features are unchanged.

3. PDF MODELING

Given a “dense” feature space—i.e., the multi-dimensional feature vector is known at each sample point of the original image—the next step in our experiments is to model the probability density function (PDF) of each feature vector parametrically. Since the exact distribution, or even the shape, is not known a priori, the following conditions are imposed: 1) In each region, it is assumed that the distribution of a particular feature can be approximated by a sum of (one or more) Gaussian PDF’s. Of course, there is no guarantee that for every image sequence each and every feature will exhibit this behavior. For the purpose of these experiments, it was found that for many “natural” image sequences most of the features' distributions could be approximated by either a unimodal or multi-modal Gaussian distribution. 2) The second condition imposed is that the user will supply a set of training points which will serve as the basis of approximating the PDF. The acquisition of these data points, or “training data” is obtained by a simple drag of the mouse over representative points from each desired region. In this manner the user implicitly defines the number of regions or classes in the sequence, while simultaneously supplying a strong clue to the shape of the PDF for each feature and region, since the feature values are known at every point in the image sequence. A major benefit to this user interaction is that the final segmentation can be made application dependent.

The difficult step of the PDF modeling task is in deciding how many modes the mixture model has. From information theory, for a given training data set of \( N \) samples, the best bi-modal distribution will fit the training data better than the best unimodal distribution. Similarly the optimal tri-modal mixture model will fit the data better than the optimal bi-modal. In fact it is clear from information theory \([12]\) that if an optimal \( N \)-modal mixture model is used, it will fit the training data precisely. The problem, however, is in determining the underlying PDF, for each feature, of all the data points (of a particular region) including both the “labeled” training data and the “unlabeled” data. In our experiments typical size for the length of training data is on the order of 100-200 points while the remaining unclassified points can be on the order of hundreds of thousands to millions. This is certainly no simple task.

The PDF modeling algorithm penalizes the models with a higher number of modes. In particular, since the objective is to determine the underlying shape of the PDF, it is important not to over fit the model exactly to the training data. The first step of the penalty function is based on empirical evidence that most regions will not exhibit more than a handful of modes. By limiting the potential number of modes, \( m \), and varying \( m \) from 1 to 5, for example, it is straightforward to calculate the parameters of the best fit to the training data, for each \( m \)-modal mixture model, using an Expectation-Maximization (EM) algorithm, \([13]\). Given a set of parametric distribution models, the next step is to measure the entropy distance, (aka. the Kullback-Leibler distance) between each of the models and the training data. As mentioned earlier, the model with the higher number of modes will be a better fit to the training data, and therefore result in a minimum entropy distance. The probability that model \( i \) is the correct model (where model \( i \) corresponds with the PDF distribution having \( i \) modes) is estimated by normalizing the derivative of the difference of entropy distances of two successive modes, with respect to \( m \) the number of modes, \([14]\). In other words, if the PDF mixture model corresponding to 2 modes brings us dramatically closer to the training data than the 1-mode model, then there is a proportionally dramatic increase in the likelihood that the 2-mode model is indeed the correct model for our purposes, even though the 3-mode model will fit the data even better than the 2-mode model. It is the rate of change, with respect to increasing number of modes, in the entropy distance that is most important than the actual entropy distance.

4. CLASSIFICATION

Given a multidimensional probability density function estimate of the feature space at each region, the remaining problem is to classify each of the remaining points to one of the \( N \) user specified regions. This type of problem is often referred to as maximum a posteriori MAP hypothesis testing, \([7]\). At each unlabeled sample point of the image we want to assign the sample to class or region \( R_i \) that maximizes the a posteriori probability, \( P(R_i | f) \), where \( f \) is the feature vector, e.g., motion, color, texture, and position value at the corresponding
location. Using Bayes' rule, this means maximizing:

\[
P(R_d | f) = \frac{P(f | R_d) P(R_d)}{P(f)}
\]

Which is the same as maximizing \(P(f | R_d)\) since the denominator does not depend on \(f\), and since there are equal priors on the region probabilities, \(P(R_d)\). Assuming the features are independent, this calculation is simply a joint product of the one-dimensional PDF, evaluated at the feature value, for each of the regions 1 through \(N\). The sample is assigned to the region corresponding with the largest product.

In order to classify the unlabeled points in successive frames, one of two methods can be used: 1) The PDF estimate is made using the training points from the first frame only, or 2) The PDF estimate is updated at each frame by tracking the training points from frame to frame using a block matching algorithm or some equivalent motion estimation.

5. SIMULATION RESULTS

Figure 6 shows two different image sequences that were segmented given a small amount of user-supplied training points. The top left image in figure 6 shows the first frame of the "Table Tennis" sequence. Overlaid on the original are the user supplied training points, where each of 5 separate regions is represented as a different gray scale value. The top right image of figure 6 shows the resulting segmentation at frames 1, 3, 6, and 10. The features used for this segmentation include position, color, luminance, texture, motion, and background difference. The bottom left image of figure 6 shows the resulting segmentation of frames 1, 21, 40, and 60.

6. ACKNOWLEDGMENTS

The research described in this paper has been supported by the Television of Tomorrow Consortium at the MIT Media Laboratory.

7. REFERENCES

Figure 1: Top Left: First frame from original “Table Tennis” image sequence with user selected training data points from 5 regions superimposed. Top Right: Segmentation result of frames 1, 3, 6, and 10, shown at half resolution, using training data from first frame only. Above Left: First frame from original “Dance” sequence. Above Right: Location of user selected training data points from 4 regions (black points are unlabeled). Left: Segmentation result of frames 1, 21, 40, and 60, shown at half resolution, using training data from first frame only.
