Object Based Segmentation of Video Using Color, Motion and Spatial Information

Sohaib Khan Mubarak Shah

Computer Vision Laboratory
School of EECS, University of Central Florida
Orlando, FL 32816
{khan,shah}@cs.ucf.edu

Abstract

Video segmentation is different from segmentation of a single image. While several correct solutions may exist for segmenting a single image, there needs to be a consistency among segmentations of each frame for video segmentation. Previous approaches of video segmentation concentrate on motion, or combine motion and color information in a batch fashion. We propose a maximum a posteriori probability (MAP) framework that uses multiple cues, like spatial location, color and motion, for segmentation. We assign weights to color and motion terms, which are adjusted at every pixel, based on a confidence measure of each feature. We also discuss the appropriate modeling of pdfs of each feature of a region. The correct modeling of the spatial pdf imposes temporal consistency among segments in consecutive frames. This approach unifies the strengths of both color segmentation and motion segmentation in one framework, and shows good results on videos that are not suited for either of these approaches.

1 Introduction

Motion information has been used for video compression for a long time. Codecs based on MPEG 1-2 and H.26x series of standards compute the motion of image blocks in a series of images and transmit only this motion information and the error in reconstructed image. If the estimates of motion are reasonably correct, the entropy of the error image is much lower than the original image, thus achieving compression. Generally, some predetermined and fixed block size is used.

Object-based segmentation, on the other hand, uses regions based on real world objects as compression primitives, rather than, say 8x8 blocks. Object based coding often creates primitives that are more homogeneous in texture and thus results in more compression. More importantly, however, it allows the use of layers in coding. Real world scenes may be considered as a rendering of views of multiple objects placed at appropriate locations. These objects may be in motion with respect to each other. If individual objects are segmented out during the coding phase, then they may be transmitted only once and the relative displacement of each layer may be transmitted in successive frames.

The key bottleneck in object-based compression is reliable segmentation of objects in an image. Image segmentation is a well studied but ill posed problem. Given a single image, there can be several 'correct' segmentations of it. Moreover, there can be several levels of segmentation. Looking at an outdoor scene, a clump of trees might be considered one segment, individual trees might be considered different segments, or individuals features of a tree (like branches, trunk, fruit) might be segmented out. It is easy to visualize cases where, by zooming the camera into the scene, our understanding of 'appropriate' segmentation might change.

Since segmentation is an ill-posed problem, the notion of 'correct' segmentation is dependent on the application. For example, for object-based compression purposes, we want to extract the background as one segment and all independently moving objects as separate segments.

The need to impose temporal consistency constraints makes video segmentation different from a series of single image segmentations. Single image segmentation can yield very different results for two very similar images. Video segmentation demands that for a given image, the segmentation achieved should relate to the segmentation of the previous image, provided that they belong to the same shot.

Spatial segmentation, in general, is based on finding regions in an image that are homogeneous with respect to some feature, while they differ significantly from other regions. Some of the most popular features used for segmentation are motion, color, texture and geometric properties. Motion segmentation is a promising paradigm for compression applications, as discussed above. Color segmentation mostly yields segments that are a superset of motion segments, i.e. color segments can be grouped together to yield motion segments. However, this is not true in all cases. It is easy to visualize a situation of a foreground object moving across a stationary background object of fairly similar color (Figure 1). In such a situation, the segments yielded by color segmentation can be such that they cannot be recreated by simply splitting motion segments.

0-7695-1272-0/01 $10.00 © 2001 IEEE

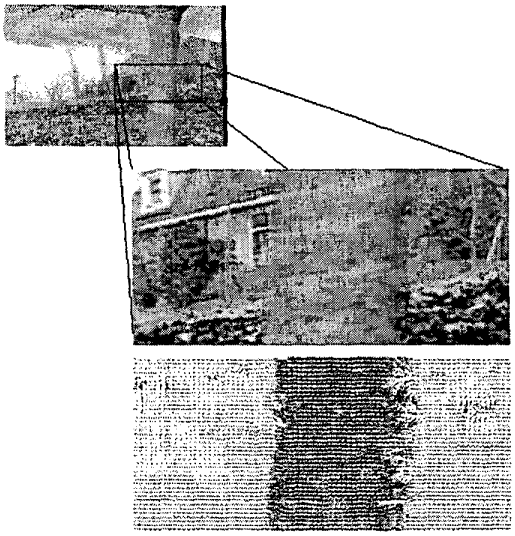

Figure 1: Color Similarity between Foreground and Background: A portion of the tree trunk matches the background. However, optical flow can be used to discriminate between them.

Motion segmentation, however, has its own problems. Due to occlusion and disocclusion, optical flow is not reliable at the boundaries of moving objects. Therefore, segmentation based on motion alone results in segments with inaccurate boundaries. Moreover, as we have stated earlier, temporal consistency is essential for correct video segmentation. Frequently objects have non-uniform motion. For example, a table tennis player will be in rapid motion for a few frames, and then may be completely still for a few frames. Using motion segmentation alone, we will be able to segment the person from the background for the frames in which he is moving, and then loose the segment for the frames in which he is still. This is not desirable for compression applications, because we do not want to update the background mosaic every few frames

To overcome this problem with motion segmentation, several researchers have proposed the use of multiple cues in the segmentation (see next section on literature review). Most previous work on the use of multiple cues involves a sequential application of features. For example an image may be segmented based on motion information, and then this segmentation may be improved based on color. In this work we propose a simultaneous combination of cues within a maximum likelihood framework, in which the weight of each feature is varied for every pixel. The choice of weights is such that they attempt to correct the two errors of motion segmentation, namely, erroneous flow at the occlusion boundaries and inconsistency in object motion. We use color and flow features and combine their strengths by adjusting these weights. Moreover, temporal consistency is not emphasized in previous work. We impose temporal consistency constraints through the use of spatial probability density functions (PDFs) to bias the maximum likelihood equation. This formulation is described in the next section.

1.1 Related Work

The idea of segmenting an image into layers was introduced by Darrell and Pentland, and Wang and Adelson [1, 8]. Darrel and Pentland [1] used a robust estimation method to iteratively estimate the number of layers and the pixel assignments to each layer. They show examples with range images and with optical flow. Wang and Adelson [8] is the seminal paper on segmenting video into layers. Affine model is fitted to blocks of optical flow, followed by a K-means clustering of these affine parameters. This step involves splitting and merging of layers, and therefore k is not fixed. After the first iteration, the shape of regions is not confined to aggregate of blocks but is taken to a pixel level within the blocks. The results presented are convincing, though the edges of segments are not very accurate, most likely due to the errors in the computation of optical flow at occlusion boundaries.

Bergen *et. al.* [3] presents a method for motion segmentation by computing first the global parametric motion of the entire frame, and then finding the segments that do not fit the global motion model well. Irani and Peleg [10] incorporate temporal integration in this loop. A weighted aggregate of a number of frames is used to register the current image with the previous one. The object that is currently being compensated for thus becomes sharply into focus and everything else blurs out, improving the stability of the solution.

The earliest work, to our knowledge, on combining multiple features for segmentation is reported by Thompson [13]. The image is segmented based on intensity and motion, by finding 4-connected regions that have similar gray-scale and optical-flow values. The regions are then merged together using a variety of heuristics. Haynes and Jain [12] attempt to find edges that are moving. The product of consecutive-frame difference picture and Sobel edge-detector output extracts moving edges.

Black [5] presents an approach of combining intensity and motion for segmentation of image sequences. Their approach is based on Markov Random Fields. They have three energy terms, of intensity, boundary and motion. Tekalp and others [16] present a system in which both color and motion segmentation is done separately, followed by clustering the color segments together that belong to the same motion segment. This assumes that the color segments are more detailed, but nevertheless accurate, than the motion segments, and only need to be grouped together for correct segmentation. We have shown earlier that this is not always the case. Another method of combining color and flow information is presented by Yang [15]. The image is first segmented using color using a multiscale segmentation formulation. Next, correspondence is established between segments to find an affine transform between the pixel location so the corresponding regions, using their centroid as

the origin. The resulting affine transforms are used to compute the smoothed optical flow at each pixel. Adjacent color segments that were initially computed are then merged together if their overall optical flow is similar. Geometric and color constraints are also applied for tracking hands.

Finally recent paper by Bergen and Meyer [4] discusses a unique approach of using both color and motion information to obtain object based segmentation. The authors argue that there is more error in optical flow at occluded side than the occluding side. Therefore, the image is segmented based on color and motion information independently, and the segment boundaries are compared with each other. The motion boundaries that do not agree with color boundaries indicate a disoccluding edge. Such analysis leads to a depth relationship between all segments which can then be grouped.

Our approach of combining multiple features for segmentation involves finding of weights of each feature at a pixel. This idea has parallels in vision literature, which might be explored for further insights. For example, there has been a large body of work in vision on sensor fusion [7]. We see the similarity of our approach with this area when we consider that each feature that we are using can be considered essentially as coming from a different sensor.

In addition to this, there has been work in the theory of integration. Poggio, Gamble and Little [11] present a theoretical framework for parallel integration of vision modules, essentially using Markov Random Fields to integrate several visual cues. Also, there has been considerable work in computer vision in the area of 3D motion segmentation. This idea follows from the structure from motion (SFM) formulations, where segmentation is done either after computing structure [2] or segmentation and computation of structure are done simultaneously [14].

2 Maximum Likelihood Estimate

Image segmentation can be considered as a classification problem. Given a set of data points (the intensity values in an image) we want to find the clustering of these data points that best corresponds to some property of these clusters. The property of interest is mostly based on minimizing the variance of some features within segments while maximizing the variance of these features across segments. That is, a segmentation is considered good if classes computed are homogeneous with respect to the feature vector, with the additional constraint that no two classes should be similar (otherwise their pixels might be grouped together into the same class).

Consider an image I_t that is to be segmented, where t denotes the time index. Suppose we want to segment this image into n classes, denoted by c_i for $i = 1$ to n. For every pixel $I_t(x,y)$, let there be a an m-dimensional *feature vector* $\mathbf{x} = [x_1, x_2, ..., x_m]^T$ which can be measured for each pixel. Let us assume that we know the class assignments of each pixel in I_{t-1}, denoted by L_{t-1}, i.e. for all pixels $I_{t-1}(x,y)$ we are given $L_{t-1}(x,y)$ such that $1 \leq L_{t-1}(x,y) \leq m$. Then we want to compute the class assignments of every pixel in frame t, i.e. L_t based on the classes in the previous frame and the new observations of the feature vector of each pixel.

Let the probability of a pixel (x,y) in frame t belonging to class c_i $(1 \leq i \leq n)$ be given by $P(c_i|\mathbf{x}_t(x,y))$, where $\mathbf{x}_t(x,y)$ is the feature vector of pixel (x,y) at time t. According to Bayes Rule:

$$P(c_i|\mathbf{x}_t(x,y)) = \frac{P(\mathbf{x}_t(x,y)|c_i)P(c_i)}{P(\mathbf{x}_t(x,y))} \quad (1)$$

Equation 1 relates the a posteriori probability $P(c_i|\mathbf{x}_t(x,y))$ to the product of $P(\mathbf{x}_t(x,y)|c_i)$ and $P(c_i)$ with the scale factor $(P(\mathbf{x}_t(x,y)))^{-1}$. We may use this equation to the find the probability of a pixel belonging to each of the classes c_1 to c_n and then assign this pixel to the class that returns the maximum probability. Thus:

$$L_t(x,y) = \arg\max_i \left\{ \frac{P(\mathbf{x}_t(x,y)|c_i)P(c_i)}{P(\mathbf{x}_t(x,y))} \right\} \quad (2)$$

where $1 \leq i \leq n$. Since the denominator does not depend on i and is always positive, we may ignore it. Also, we can multiply with log function, to get the log likelihood relationship:

$$L_t(x,y) = \arg\max_i \{\ln(P(\mathbf{x}_t(x,y)|c_i)) + \ln(P(c_i))\} \quad (3)$$

The prior term in Equation 3 may be ignored if all classes are equally likely to be observed. The a priori term $P(\mathbf{x}_t(x,y))$ can be computed for a given observation if the *probability density function* (pdf) of the form $P(\mathbf{x}|c)$ is known. If we make some reasonable assumptions about this m-dimensional pdf, then we may compute L_t.

We assume that the given video is already segmented into shots. The first frame of each of these shots is segmented using some clustering scheme. This step is called initial segmentation, and is described later. For now, we assume that the previous frame is already segmented. For the next frame, the segments are computed using the information about the segments in the previous frame.

To impose temporal consistency, we also use the spatial location and spread of each segment as a feature. This biases our solution such that we are more likely to pick a solution with least change in location of segments.

Thus our 7-dimensional feature vector is given by $\mathbf{x} = [x, y, Y, U, V, u, v]^T$, where (x,y) denotes the spatial location, $[Y,U,V]$ denotes the color and $[u,v]$ denotes the optical flow at the pixel. All these values can be measured at each pixel. The location (x,y) and the color (Y,U,V) is given directly by reading the image file. The flow component is computed using a hierarchical version of Lucas-Kanade's optical flow method [9].

We assume that the three sets of features in $\mathbf{x}$ are mutually independent, i.e. their covariance matrix may be written as a block diagonal, with zero off-diagonal terms. This

assumption is valid because given the spatial location of a segment, we cannot predict its color or optical flow. Similar, a particular color may occur anywhere in the image and that segment might be moving with any velocity. Thus

$$\Sigma = \begin{bmatrix} \sigma_{xx}^2 & \sigma_{xy}^2 & 0 & 0 & 0 & 0 & 0 \\ \sigma_{yx}^2 & \sigma_{yy}^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_{YY}^2 & \sigma_{YU}^2 & \sigma_{YV}^2 & 0 & 0 \\ 0 & 0 & \sigma_{UY}^2 & \sigma_{UU}^2 & \sigma_{UV}^2 & 0 & 0 \\ 0 & 0 & \sigma_{VY}^2 & \sigma_{VU}^2 & \sigma_{VV}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_{uu}^2 & \sigma_{uv}^2 \\ 0 & 0 & 0 & 0 & 0 & \sigma_{vu}^2 & \sigma_{vv}^2 \end{bmatrix} \quad (4)$$

With this assumption, the probability $P(\mathbf{x}_t(x,y)|c_i)$ can be broken down into a product of three probability terms. It is important to model the pdfs of these probabilities reasonably well. We assume a Gaussian pdf for the color and flow components. This assumption indicates normally distributed noise around the color or flow mean of a class.

$$P(\mathbf{x}_t(x,y)|c_i) = \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma_i|^{\frac{1}{2}}} e^{-\frac{(\mathbf{x}_t - \mu_i)^T \Sigma_i^{-1} (\mathbf{x}_t - \mu_i)}{2}} \quad (5)$$

In this equation, d is 2 for flow pdf and 3 for color pdf. μ_i and Σ_i refer to the mean and the covariance matrix of the feature vectors of all the pixels belonging to segment c_i. We use this equation to find the color and flow likelihood of every pixel belonging to each class c_i. The mean of class c_i at time t, $\mu_{i,t}$, and the covariance matrix $\Sigma_{i,t}$ should be known. We can compute $\mu_{i,t-1}$ and $\Sigma_{i,t-1}$ from the given segments in frame $t-1$. For this implementation, we assume that predicted pdf at frame t, $\hat{P}(\mathbf{x}_t(x,y)|c_i)$ is the same as the computed pdf at frame $t-1$, $P(\mathbf{x}_{t-1}(x,y)|c_i)$. It is easy to modify this implementation so that the spatial, color and flow components of the mean vector have a model of how they change from the computed position in the previous frame to the predicted position in the new frame.

For the spatial term, however, a Gaussian pdf is not a reasonable assumption. Segments in the image can be of any arbitrary shape and not necessarily elliptical. Thus, we use a pdf estimated from the actual data, rather than a parametric model. We follow the approach of [6] for estimating the spatial pdf of each class k:

$$P_s(\mathbf{X}) = \frac{m^{-1}}{(2\pi)^{\frac{d}{2}}\sigma^d} \sum_{i=1}^{m} \exp\left\{-\frac{(\mathbf{X} - \mathbf{X}_{ki})^T (\mathbf{X} - \mathbf{X}_{ki})}{2\sigma^2}\right\} \quad (6)$$

where $\mathbf{X}_{ki}$ is the ith data point from class k, m is the total number of data points, σ is a smoothing parameter and d is the dimensionality of the space (in the case of spatial pdf, it is 2). This pdf is essentially generated by generating a Gaussian distribution for every data point as its mean, and adding all these distributions together. The smoothing parameter σ governs the width and the smoothness of the resulting pdf. This pdf can be efficiently implemented by convolving the binary mask of a class with a Gaussian kernel. The size of the kernel can be limited by the value of σ, so that very small values are treated as zero. The value of σ effectively limits the amount of motion allowed between two consecutive frames, and if large movement is expected within two frames (due to fast objects close to the camera or low frame-rate) a larger σ should be used.

Putting these terms together in the log-likelihood equation (3) yields:

$$L_t(x,y) = \arg\max_i \{\ln(P_s(\mathbf{X})) + \ln(P(\mathbf{x}_t^c(x,y)|c_i)) + \ln(P(\mathbf{x}_t^f(x,y)|c_i))\} \quad (7)$$

Since we have written the space, color and flow terms separately, we can assign individual weights to each of the terms, if one set of features needs to be emphasized more than the others. Let the space, color and flow components on RHS of Equation 7 be denoted by $L_t^s(x,y,i)$, $L_t^c(x,y,i)$, $L_t^f(x,y,i)$ respectively. Then, by adding a separate weight for each term at every pixel, we may rewrite our likelihood equation as:

$$L_t(x,y) = \arg\max_i \{w_t^s(x,y)L_t^s(x,y,i) + w_t^c(x,y)L_t^c(x,y,i) + w_t^f(x,y)L_t^f(x,y,i) \quad \text{for} 1 \le i \le n \quad (8$$

Equation 8 represents the final form of the maximum likelihood estimate for segmenting video.

3 Computing Feature Weights

Overall results of the segmentation will heavily depend on the weights. We use individual weights for each set of features for every pixel in a frame (Equation 8). We have developed a set of heuristics for selecting these weights, which are based on our understanding about the desired segmentation. The basic paradigm behind the selection of these weights follows from our discussion about the shortcomings of motion segmentation in Section 1. We have observed that motion segmentation should work better than color segmentation if we can overcome the problem of errors at boundaries due to unreliable optical flow and temporal inconsistency. It is this shortcoming of motion segmentation that dictates our choice of weights.

We compute optical flow by a hierarchical version of Lucas-Kanade method [9]. The optical flow is smooth and reliable within objects, but is erratic and unreliable at the object boundaries. This is so because during occlusion, the nature of the patch to be matched changes within a frame as some pixels get occluded. Thus, the we may pick an arbitrary match as the flow output.

We propose two steps to remedy the errors of motion segmentation. We fix w^s (spatial) to 1, and balance between the optical flow and color terms. To correct for errors at occlusion boundaries, If flow is 'reliable', we weight the flow term more, and correspondingly decrease the weight of the color term, and vice versa. To compute the reliability of flow, we look at the value of the flow likelihood $L_t^f(x,y,i)$. This term returns the probability of the match of optical flow

at a pixel with all the classes. The maximum value of this term gives the probability of the best match. The higher this probability, the better this pixel matches to one of the existing classes. For noisy flow areas, this probability is low, indicating the we should give a higher weight to the color term in those areas.

We normalize the log likelihood $L_t^f(x,y,i)$ between 0-1 by multiplying it with a sigmoid function. This yields our confidence measure ρ_1.

$$\rho_1 = \left\{1 + exp(-a(\max_i(L_t^f(x,y,i)))\right\} \quad (9)$$

where a is a positive number defining the slope of the sigmoid function. We use $a = 0.5$ in all our experiments.

The second error in motion segmentation arises out of inconsistency in motion. If an object moves with respect to the background but then stops, it should not be immediately made the part of the background, but should still be segmented out. We cater for this effect by computing the absolute difference between the maximum value of $L_t^f(x,y)$ and the second maximum value from amongst the significant neighbors of this class. If the difference in their probabilities is low, then this means that optical flow is not providing enough discriminatory information. In such a scenario, we want to weigh color more. Thus we find

$$d = |max(L_t^f(x,y)) - max2(L_t'^f(x,y))| \quad (10)$$

where $max2$ is the highest probability value returned from the neighbors of the class which returned the highest probability value. Since d is always positive, we normalize it between 0-1 by multiplying it with a shifted sigmoid function

$$\rho_2 = \{1 + exp(-a(d-t))\} \quad (11)$$

where t is an appropriate shift parameter for the sigmoid function.

Having computed ρ_1 and ρ_2, we find the weights w^f and w^c as compliments of each other.

$$w^f = \rho_1\rho_2 \quad (12)$$

$$w^c = 1 - (\rho_1\rho_2) \quad (13)$$

3.1 Initial Segmentation

We segment the first frame by a two step process. The first step to apply the EM algorithm on color and optical flow data of the image to essentially find a Gaussian mixture model that fits this 5D data. For each pixel, we iteratively compute its likelihood of belonging to one of the Gaussian distributions, and then the new parameters of the distribution. We perform a fixed number of iterations of this process. The next step is to take the regions generated by this mixture model and use them to compute classes in the likelihood equation (Equation 7) to compute the new regions. This process is repeated a number of times for the same frame, till the regions are stable. The first step generates seed regions which are refined through the second step, which incorporates both the idea of flow weights and the spatial pdf. The classes thus obtained are suitable to be used for the next frame.

4 Experiments

In our experiments, we have used two test sequences that present different challenges to the video segmentation problem. The flower sequence contains objects moving only due to camera motion, and thus their motion is smooth and consistent. Portions of the image in this sequence consist of small texture, with a number of colors in it. Thus, color segmentation does not yield satisfactory results for object based compression. Motion segmentation of this sequence yields errors too, at the boundaries of the tree. The second test sequence is the table-tennis sequence. Its characteristics are almost entirely opposite to the flower sequence. There is almost zero background motion in the first part of the sequence. The texture is largely uniform and smooth, and therefore color segmentation alone does a decent job of extracting the objects of interest. Motion segmentation, on this sequence, suffers from temporal consistency problems, because the player repeatedly moves and stops. Therefore the player is very visible in the magnitude of optical flow for a few frames and then disappears for a few frames. Results are presented sequences using the method described in this paper. It can be seen that the method works well to correct both the errors mentioned above. It corrected the occlusion boundaries errors in flower sequence, so that the tree segment has accurate boundaries. It also keeps the player segmented out even when he stops and has the same optical flow as the background for a number of frames.

In the flower sequence, the motion of the objects is smooth and continuous, therefore, the objects are always segmented out well using motion information. In the tennis sequence, however, motion information fails once the player stops. In such cases, the color representation of the segments takes over, as shown in Figure 3, which results in greater temporal consistency and better segmentation than motion alone. Moreover, the segmentation is more correct in defining the object, compared to color segmentation. Notice that since we started with a clustering of motion vectors as our initial segmentation, the first frame consisted of a bad segmentation, but that was readily corrected within a few frames. The algorithm performed well for this sequence, for an object-based segmentation application. Movies of results, and results on more sequences are available at http://www.cs.ucf.edu/~khan/research.html

5 Summary

We have proposed a maximum likelihood framework for video segmentation. By combining multiple cues of motion and color at each pixel, our method yields meaningful segments. Specifically, the boundaries of our segments are accurate and we achieve better temporal consistency.

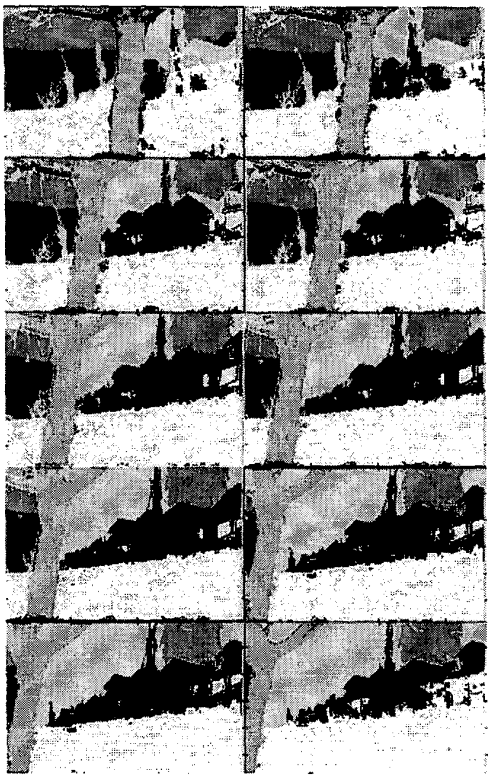

Figure 2: Segmentation of Flower Sequence using both color and motion information. Each frame is segmented into 6 classes. Notice the edges of the tree are correctly segmented. Every 4th frame is shown

Figure 3: Segmentation of Tennis Sequence using both color and motion information. Each frame is segmented into 6 classes. Notice that the player has inconsistent motion, and is often stationary, matching with the background. However, color based segmentation takes over in this situation. Every 10th frame is shown

References

[1] Alex Pentland A. Darrel. Cooperative robust estimation using layers of support. T.R. 163, MIT Media Lab, Vision and Modeling Group, Feb 1991.

[2] Gilad Adiv. Determining three-dimensional motion and structure from optical flow generated by several moving objects. *PAMI*, 7(4):384–401, July 1985.

[3] J.R. Bergen, P. Anandan, K. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *ECCV*, pages 237–252, May 1992.

[4] L. Bergen and F. Meyer. "a novel approach to depth ordering in monocular image sequences". In *CVPR*, pages 536–541, June 2000.

[5] Michael Black. Combining intensity and motion for incremental segmentation and tracking over long image sequences. In *ECCV*, 1992.

[6] T. Cacoullos. Estimation of a multi-variate density. *Annals of Institute of Statistical Mathematics, Tokyo*, 18(2):179–189, 1966.

[7] Jay K. Hackett and Mubarak Shah. *Trends in Optical Engineering*, chapter Multi-sensor Fusion: A Perspective. 1993.

[8] Edward H. Adelson John Y. A. Wang. Representing moving images with layers. *IEEE Transactions on Image Processing*, September 1994.

[9] B. D. Lucas and T. Kanade. An iterative image registration technique with application to stereo vision. In *Proc. 7th International Joint Conference on Artificial Intelligence*, pages 674–679, 1981.

[10] Shmuel Peleg Michal Irani. Motion analysis for image enhancement: Resolution, occlusion and transparency. *Journal of Visual Communication and Image Representation*, 4(4):324–335, December 1993.

[11] T. Poggio, E. B. Gamble, and J. J. Little. Parallel integration of vision modules. *Science*, 242:436–440, 1988.

[12] Ramesh Jain Susan M. Haynes. Detection of moving edges. *CVGIP*, 21:345–367, 1983.

[13] William B. Thompson. Combining motion and contrast for segmentation. *PAMI*, pages 543–549, Nov 1980.

[14] Yu Tian and Mubarak Shah. Motion estimation and segmentation. *Machine Vision and Applications*, 9:32–42, 1995.

[15] Ming-Hsuan Yang. *Hand Gesture Recognition and Face Detection in Image*. Ph.d. thesis, University of Illinois at Urbana-Champange, 2000.

[16] A. Murat Tekalp Yucel Altunbasak, P. Erhan Eren. Region based parametric motion segmentation using color information. *Journal of Graphical Models and Image Processing*, 60(1):13–23, January 1998.