Efficient Temporal Mean Shift for Activity Recognition in Video

Yan Ke[1], Rahul Sukthankar[2,1], Martial Hebert[1]
[1]School of Computer Science, Carnegie Mellon; [2]Intel Research Pittsburgh
{yke,rahuls,hebert}@cs.cmu.edu

1. Introduction

We propose an temporal mean shift algorithm that clusters spatio-temporal regions in video by exploiting the temporal nature of video. Extracting spatio-temporal regions is often one of the first pre-processing steps in an activity recognition system [3, 6, 8]. Due to the high volume of data present in video, higher level algorithms are often unable to do inference directly at the pixel level. A spatio-temporal segmentation algorithm that extracts coherent regions in apperance and motion can reduce the amount of data that needs to be processed by higher level inference algorithms. Thus, it is essential to have an efficient segmentation algorithm that can quickly process large amounts of video data. In our previous work in activity recognition, we introduced a set of volumetric features for efficiently classifying spatio-temporal regions in video [6]. One limitation of this work is that each action had to be manually segmented and labelled. A clustering algorithm that can automatically extract spatio-temporal regions will allow semi-supervised training of our classifier and allow us to recognize a wider range of actions. We propose to use mean shift as a basis for our clustering algorithm. Despite all of the work in improving standard mean shift, described below, it is still too slow for practical use on video.

Mean shift was first used for image segmentation by Comaniciu *et al.* [1]. Since then, mean shift has gained wide-spread popularity as a general clustering and segmentation algorithm in the vision community. Georgescu *et al.* improves on standard mean shift by using locality sensitive hashing [5] for the near-neighbor search [4]. Wang *et al.* adapts the kernel used in mean shift to be anisotropic and achieves better results in segmenting video [10]. Other efforts in segmenting regions in video include [2, 7, 9, 11].

2. Approach

Our key contribution is the insight that mean shift clustering can exploit the fact that there is typically very little change between successive video frames. Most of the pixels, and therefore the clusters, shift only slightly from frame to frame. Since mean shift is an iterative procedure, fewer iterations are required for convergence if the initial search is already close to the local optimum. Our temporal mean shift algorithm exploits the the temporal similarity between successive frames by initializing the search using the modes found in the previous frame.

We now describe how we can use standard mean shift to extract spatio-temporal regions. Then, we introduce an temporal mean shift algorithm that makes it possible to do efficient clustering on long sequences of video. Mean shift is an iterative algorithm that can be used to find the local mode of a cluster of points [1]. Each pixel i at location (x_i, y_i) and time t has colors (r_i, g_i, b_i). From this information, we need to calculate the local appearance of the pixel and its motion path. The local appearance could be in the form of texture features, calculated using Gabor filters, and its raw color values. The motion is calculated using dense optical flow, for example using Lucas-Kanade. Each pixel is then transformed into a high dimensional feature vector that contains its appearance and motion information. Let f_i be a d-dimensional point in a set of n feature points, $f_1 \dots f_n$. To cluster the points, we find the mode near f_i using mean shift. We calculate

$$y_{(i,1)} = f_i \tag{1}$$

$$y_{(i,j+1)} = \frac{\sum_{k=1}^{n} f_i g(||\frac{y_{(i,j)} - f_k}{h}||^2)}{\sum_{k=1}^{n} g(||\frac{y_{(i,j)} - f_k}{h}||^2)}, \tag{2}$$

where $j = 1, 2, \dots$, with kernel g, typically a Gaussian kernel, and bandwidth h. The mode y_i is the limit of the series, where $y_{(i,j)}$ converges to a fixed point. Each mode represents a spatio-temporal cluster that has similar appearance and motion. The main difficulty in using standard mean shift is that it is computationally intensive. The inner loop of mean shift requires a range-search of neighbors f_k near $y_{(i,j)}$, which is particularly time-consuming in high dimensions. Although there has been considerable efforts in optimizing mean shift, a 640×480 image could take up to a minute to segment [4] — and much longer to segment an entire video clip.

Now we sketch our proposed temporal mean shift algorithm. To optimize the inner loop of mean shift, one can either speed up the nearest-neighbor search (as in [4]), or reduce the number of iterations required to converge to the local mode (as in our approach). Figure 1 shows a person sitting down, captured at 30 frames per second.

Figure 1: Part of a video capturing a person sitting down. Ignoring the static background, the difference between successive frames is small because the video is captured at thirty frames a second. Therefore, the cluster centers found in frame t can be used to initialize the clustering process in frame $t+1$.

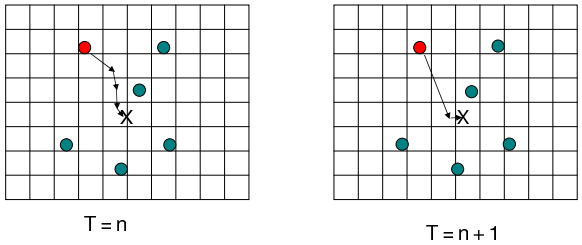

Figure 2: Using the cluster center found in frame n to initialize the clustering in frame $n+1$. Starting at upper left point, it takes four iterations to converge to the mode at point X at frame n. At frame $n+1$, all of the features shift to the right slightly, and we are able to find the new mode in just two iterations by starting at the mode found in the previous frame.

Discounting the stationary background, the difference between each frame is very small. Therefore, the feature points in frame t are likely to be close to the feature points in frame $t-1$. Similarly, the cluster centers for one frame should be close the those of the previous frame as well. More formally, let f_i^t denote a feature point at pixel (x_i, y_i, t), and let y_i^t denote the mode found using mean shift, starting at at (x_i, y_i, t). We expect $|f_i^{t+1} - f_i^t|$ to be small. This will cause $|y_i^{t+1} - y_i^t|$ to be small as well. Therefore, by initializing $y_{(i,1)}^{t+1}$ to be y_i^t, a small number of iterations is sufficient to reach convergence to the new mode. This is illustrated in Figure 2.

3. Results

We compare our temporal mean shift algorithm against the classical mean shift by running them on video similar to the one shown in Figure 1. First, we analyze the number of iterations it takes for each algorithm to converge, and then we analyze the segmentation results. Our video is 160×120 pixels in size and we have converted it to gray scale. The results shown are preliminary, and we expect to have more extensive results in the poster. Currently, we use only the x, y pixel coordinates and its intensity value, but we plan to use texture and motion features in the poster results. A bandwidth of 25 pixels and 0.04 in intensity values are used. The standard mean shift algorithm takes an average of 9.5 iterations to converge per pixel. On the other hand, our temporal mean shift algorithm takes an average of 5.0 iterations, for a speed increase of nearly 100%. The average distance in the cluster centers between the original and our algorithm is only 4.9 pixels.

4. Conclusion

We introduced an temporal mean shift algorithm that is optimized for extracting spatio-temporal regions in video. The key insight is that there is typically very little change between successive video frames. By using the cluster locations found in the previous frame to initialize the mode search for the next frame, we achieve a speedup of nearly 100% with a small error in clustering performance.

References

[1] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2002.

[2] D. Cremers and S. Soatto. Variational space-time motion segmentation. In *Proceedings of International Conference on Computer Vision*, 2003.

[3] D. DeMenthon and D. Doermann. Video retrieval of near-duplicates using k-nearest neighbor retrieval of spatio-temporal descriptors. *Multimedia Tools and Applications*, 2005.

[4] B. Georgescu, I. Shimshoni, and P. Meer. Mean shift based clustering in high dimensions: A texture classification example. In *Proceedings of International Conference on Computer Vision*, 2003.

[5] P. Indyk and R. Motwani. Approximate nearest neighbor – towards removing the curse of dimensionality. In *Proceedings of Symposium on Theory of Computing*, 1998.

[6] Y. Ke, R. Sukthankar, and M. Hebert. Efficient visual event detection using volumetric features. In *Proceedings of International Conference on Computer Vision*, 2005.

[7] S. Khan and M. Shah. Object based segmentation of video using color, motion and spatial information. In *Proceedings of Computer Vision and Pattern Recognition*, 2001.

[8] I. Laptev and T. Lindeberg. Space-time interest points. In *Proceedings of International Conference on Computer Vision*, 2003.

[9] J. Shi and J. Malik. Motion segmentation and tracking using normalized cuts. In *Proceedings of International Conference on Computer Vision*, pages 1154–1160, 1998.

[10] J. Wang, B. Thiesson, Y. Xu, and M. Cohen. Image and video segmentation by anisotropic kernel mean shift. In *Proceedings of European Conference on Computer Vision*, 2004.

[11] J. Xiao and M. Shah. Motion layer extraction in the presence of occlusion using graph cut. In *Proceedings of Computer Vision and Pattern Recognition*, 2004.