Long-Term Moving Object Segmentation and Tracking Using Spatio-Temporal Consistency

Di Zhong and Shih-Fu Chang
{dzhong, sfchang}@ee.columbia.edu
Department of Electrical Engineering, Columbia University, NY, USA

Abstract The success of object-based media representation and description (e.g., MPEG-4 and –7) depends largely on effective object segmentation tools. In this paper, we expand our previous work on automatic video region tracking and develop a robust - moving objects detection system. In our system, we first utilize innovative methods of combining color and edge information in improving the object motion estimation results. Then we use the long-term spatio-temporal constraints to achieve reliable object tracking over long sequences. Our extensive experiments demonstrate excellent results in handling challenging cases in general domains (e.g., stock footage) including depth-varying multi-layer background and fast camera motion.

1. Introduction

The newly established MPEG-4 standard has proposed an object-based framework for efficient multimedia representation. Similarly, the upcoming MPEG-7 standard, which aims at offering a comprehensive set of audiovisual description tools, also adopts an object-oriented model to capture information about objects, events, scenes and their relationships. In both standards, segmentation of objects is non-normative and is left to technology developers and researchers. Thus, the success of object-based media representation and description depends largely on effective tools for object segmentation.

Although much work has been done in decomposing images into regions with uniform features, we are still lacking robust techniques for segmenting semantic video objects in general video sources. In our previous work, AMOS [7], we developed a general interactive tool for semantic object segmentation. It can be used in offline applications where object-based compression and indexing is needed. In the case when real-time processing is required, user inputs are usually not feasible or very limited. For example, in broadcast sports or news programs, if we want to parse and summarize video objects and events in real time, automatic object extraction methods are needed.

In this paper, we apply and expand our previous work on automatic video region tracking [6] and develop an automatic moving object tracking system by grouping low-level regions using domain models. Specifically, we will look at the motion characteristics of objects, and extract salient moving objects from complex scenes. Our main objectives are: real-time, fully automatic, and capable of handling practical situations involving complex scenes. These combined features distinguish our system from existing works.

Except for some special cases (e.g., surveillance videos), common TV programs and home videos usually contain camera motions. In these situations, to detect moving objects, we first need to compensate motions caused by camera operations. As pointed out in [1], the camera induced image motion depends on the ego-motion parameters (i.e., rotation, zoom and translation) of the camera and the depth of each point in the scene. In general, it is an inherently ambiguous problem to estimate the depth information and these physical parameters. Existing camera motion detection approaches can be generally divided into two classes: 2D algorithms that assume the scene can be approximated by a flat surface, and 3D algorithms that work well only when significant depth variations are preserved in the scene. It has been noticed [1] that in 2D scenes when the depth variations are not significant, the 3D algorithms are not robust or reliable. On the other hand, 2D algorithms using a 2D global parametric model (e.g., affine model) cannot handle 3D scenes where there are multiple moving layers under camera motions.

As typically depth information is not well preserved, 2D algorithms are used more widely than 3D algorithms. When the scene is far from the camera and/or the camera motion only includes rotation and zoom, a single affine motion model can be used to model and compensate the camera-induced motion. However, when the scene is close to the camera and the camera is translating, multiple moving planar surfaces may be produced in the image sequence. For example, in Figure 3, the fourth sequence contains many motion layers- the ground, the skater and the wall. In general, the above two scenarios may follow each other in

the same video shot with gradual transitions between them. To manage this problem, many approaches have been proposed to use multiple 2D parametric models to capture multiple motion layers.

In [5], affine motion parameters are first estimated from the optical flow by linear regression, and then spatio-temporal segmentation is obtained by a clustering in the affine parametric space. In [2], a dominant motion is first estimated by means of a merge procedure. Then motion vectors that can be well represented by this dominant motion model are identified and excluded, and secondary affine parameters are estimated from remaining blocks. This procedure is repeated until all motion layers are detected. Similar approaches are also reported in [4]. These methods rely only on motion information in grouping image pixel or blocks into motion layers, and thus usually result in inaccurate segmentation on motion boundaries. As there is a strong dependence between motion estimation and layer segmentation, without good segmentation to begin with, motion estimation results will not be accurate. Another problem in most prior works is that object tracking is not adequately addressed. It is assumed that moving objects detected at individual frames automatically form the temporal object track. In real-world scenes, objects and camera usually do not have uniform temporal motions. Objects may show obvious motion in some frames, but show slight or even no motions in other frames. This introduces inconsistent detection results in long sequences.

To solve these problems, expanding our region segmentation and tracking algorithms proposed in [6], we develop a two-stage moving objects detection method. This method uses regions with accurate boundaries to effectively improve motion estimation results, and uses the temporal constraint to achieve more reliable object tracking results over long sequences. In the rest of the paper, we will first give an overview of the system. The two detection stages are discussed in section 3 and 4. Experiment results and discussions are given in section 5.

2. System Overview

The system contains two stages (**Figure 1)**. In the first stage, we apply an iterative motion layer detection process based on the estimation and merging of affine motion models. Each iteration generates one motion layer. The difference from existing methods is that motion models are estimated from spatially segmented color regions instead of just pixels or blocks.

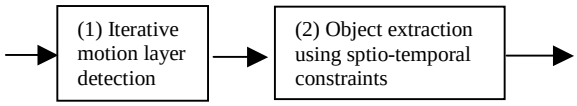

Figure 1. Two-stage moving object detection based on region segmentation and tracking

In the second stage, temporal constraints are applied to detect moving objects in spatial and temporal space. Layers in individual frames are linked together based on characteristics of their underlying regions. One or more layers will be declared as motion objects according to specific spatio-temporal consistency rules.

3. Iterative Motion Layer Detection

The iterative layer detection is applied to each individual frame as shown in **Figure 2**. The initial input to the system includes image regions automatically extracted using color and edge information. First, non-background regions[1] are merged into motion layers according their affine motion models, e.g. the 8-parameter ego-motion model. Because different regions that belong to the same motion layer may have different estimated parameters due to inaccuracy in the initial dense motion field, a simple clustering approach in the affine parametric space usually does not work well. To solve this problem, we use the following distance measure to compare two neighboring regions R_i and R_j.

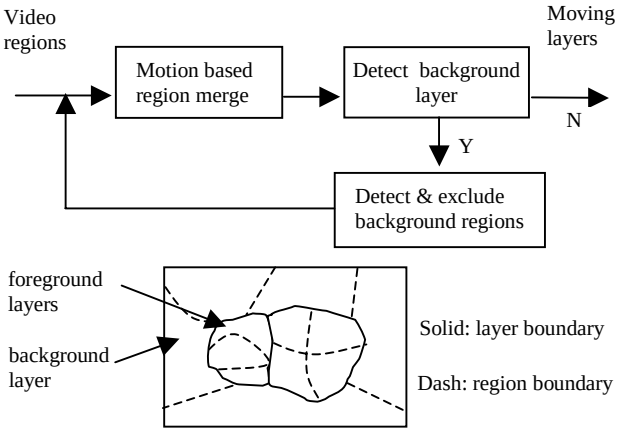

Figure 2. Iterative motion layer detection procedure

$$D(i,j) = \min(MCErr(R_i, M_j), MCErr(R_j, M_i)) \qquad (1)$$

where M_i and M_j are the affine motion models of region R_i and R_j respectively. $MCErr(R, M)$ is the motion compensation error of region R under motion model M. A region *i* is merged with its closest neighbor if their distance is below a given threshold *TH_AFF*.

After regions are merged into motion layers, we try to identify one background layer in each iteration. This is based on the assumption that a foreground layer usually has discontinued motion fields around most of its outer boundaries, while the background layer usually has continuous outer boundaries with neighboring background layers. Boundaries of a layer are consisted of pixels that

[1] In the first iteration, all regions are non-background regions

have at least one neighboring pixel not belonging to the layer. Outer boundary is the outmost closed curve that contains the whole layer. Assume $b_1,\ldots,b_n$ are the *n* points along the outer boundary of a layer *l* (do not consider pixels on the frame boundary), we define the following energy function to measure its boundary discontinuity.

$$E_l = \frac{1}{n}\sum_{p=b1}^{bn} G(p) \quad \text{and}$$

$$G(p) = \max(|p_1 - p_8|, |p_2 - p_7|, |p_3 - p_6|, |p_4 - p_5|) \qquad (2)$$

where *p1-p8* are motion vectors of *p*'s 8 neighbors (clockwise, *p1* at left-upper corner). This energy function is similar to common edge detection operators such as the Roberts operator. A layer *l* is detected as a potential background layer only when E_l is smaller than a threshold (e.g., 0.4). If no background layer is detected, the algorithm stops and all remaining regions belong to foreground layers. When there are more than one possible background layers, the largest one is chosen as the background, and its affine motion model is used to compensate non-background regions. Those regions with small compensation errors are classified as background, and excluded from the next iteration of layer merging and detection. After multiple iterations, multiple background layers may be produced, while multiple foreground layers remain.

4. Object Extraction Using Spatio-Temporal Constraints

The foreground layers detected at individual frames may be reliable. There are several reasons. First, the motion field and motion models may be inaccurate. Second, more importantly, a moving object may have noticeable motions in some frames where it can be easily detected. But in other frames, it may be static and is mistakenly treated as background. A long-term decision through a long-term interval (e.g., a shot) is necessary to remove such errors and achieve reliable results.

To apply temporal constraints, we first link foreground layers (i.e., tracking) in individual frames according to their underlying regions. A foreground layer L^m in frame *m* is linked with a layer L^n in frame *n*, if the following condition is satisfied:

$$\left|L^m \cap L^n\right| = \max_{k,l}\left(\left|L_k^m \cap L_l^n\right|\right) \qquad (3)$$

where L_k^m and L_l^n are the *k*th and *l*th foreground layer in frame *m* and *n* respectively. The maximum is computed over all foreground layers in frame *m* and *n*. The intersection of two layers in Eq (3) is defined as the number of common regions they both contain. Two regions in different frames are said to be common if one is tracked by motion projection from another one. In other words, layer L^m in frame *m* is linked to the layer in a previous frame (*n*) that shares the most common regions. This process is iterated to foreground layers remaining unlinked. In addition, we also define the link as a conductive relationship, which means if layer A and B, B and C are linked respectively, then A and C are also linked. This ensures that each local motion layer belongs to one and only one temporal layer.

The above linking or tracking process results in a number of groups of foreground layers. We will refer these groups as *temporal layers* below. We use some spatio-temporal constraints to validate these temporal layers. The first one is the duration of a temporal layer. Layers with short duration are likely to be noise or background regions, and thus are dropped. Secondly, the frame-to-frame changes of center coordinates and sizes of a temporal layer are examined. If there are large and abrupt changes, the temporal layer is not a valid tracking and will not be detected as a foreground object.

Finally, a morphological open and close procedure is applied at individual frames to remove small isolated regions and to fill holes within a moving layer. There are some issues that are not addressed in our approaches. For example, the temporal occlusion is not considered here. When one moving object is first moving, then occluded by another object or background, and later appear as a separate moving object again, it will be treated as a new moving object. However, we can use region based object matching [3] to detect reoccurrence of the same object.

5. Results and Discussion

In **Figure 3**, each row includes the image of frame #1, and then shows the moving object tracking results at frame #1, #10, #20 and #30. They all have depth variance and camera motion (i.e. following the moving objects) in the scenes, resulting in multiple motion layers. The first sequence contains a skater running towards the camera. The ice field has a gradual depth change from near to far. In the second sequence, a person is working away from the camera in an office. Cubic walls exit at different depths. The third sequence is a bird-eye's view of a soccer player running in the field. Sequence 4 contains three background layers, which are the ground, wall and crowd. The last sequence contains the sky, the stage and the jumping skier. Note that regions within segmented objects are shown in random colors to demonstrate region segmentation results. One region being tracked at different frames is shown with the same color.

The gradual depth change in the sequence 1 does not cause much problem as the ground is merged into one large region in the first color based region segmentation stage. In sequence 2, the cubic walls are tracked as separated regions. Although these regions are classified as foreground motion layers in some frames, their temporal durations are short and thus are considered as background. In the third sequence, both the player and the grass field have gradual

depth variances. Similar to the first sequence, color segmentations are proven to be useful in handling such situations. The above three sequences show good tracking results. Some small background regions are falsely included in the sequence 4. These regions are mainly from the connecting parts of two background regions, and usually have inaccurate motion fields. Some foreground pixels are missed in (5) is because small isolated regions are removed in the final morphological operations.

In summary, our experiments demonstrated that long-term region based moving object detection approach is more robust and reliable compared to existing approaches that only uses local motion information (e.g., frame-to-frame motion field). The method is designed to automatically detect and track salient moving objects within scenes with multiple motion layers. By using temporal constraint, we can robustly and accurately segment moving objects over a long period. Our method can also handle objects with discontinuous motions (i.e., moving in some frames and still in other frames).

References:

1. G. Adiv, “Inherent ambiguities in recovering 3D motion and structure from a noisy flow field”, IEEE Trans. on Pattern Analysis and Machine Intelligence, 11:447-489, May 1989.
2. G. D. Borshukov, G. Bozdagi, Y. Altunbasak and A.M. Tekalp, “Motion segmentation by multistage affine classification”, IEEE transaction on image processing, Vol 6, No 11, Nov 1997.
3. S.-F. Chang, W. Chen, H. Meng, H. Sundaram, and D. Zhong, "VideoQ: An Automated Content-Based Video Search System Using Visual Cues", ACM 5th Multimedia Conference, Seattle, WA, Nov. 1997.
4. F. Moscheni, F.Dufaux and M.Kunt, “A new two-stage global/local motion estimation based on a background/foreground segmentation”, IEEE Proc ICASSP’95, Detroit, MI, May 1995.
5. J.Y.A.Wang and E.H.Adelson, “Spatio-temporal segmentation of video data”, SPIE Proc Image and Video Processing II, San Jose, CA, Feb 1994.
6. D. Zhong and S.-F.Chang, "Video Object Model and Segmentation for Content-Based Video Indexing", ISCAS'97, HongKong, June 9-12, 1997.
7. D. Zhong and S.-Fu Chang, "AMOS - An Active MPEG-4 Video Object Segmentation System", ICIP-98, Chicago, Oct. 1998.

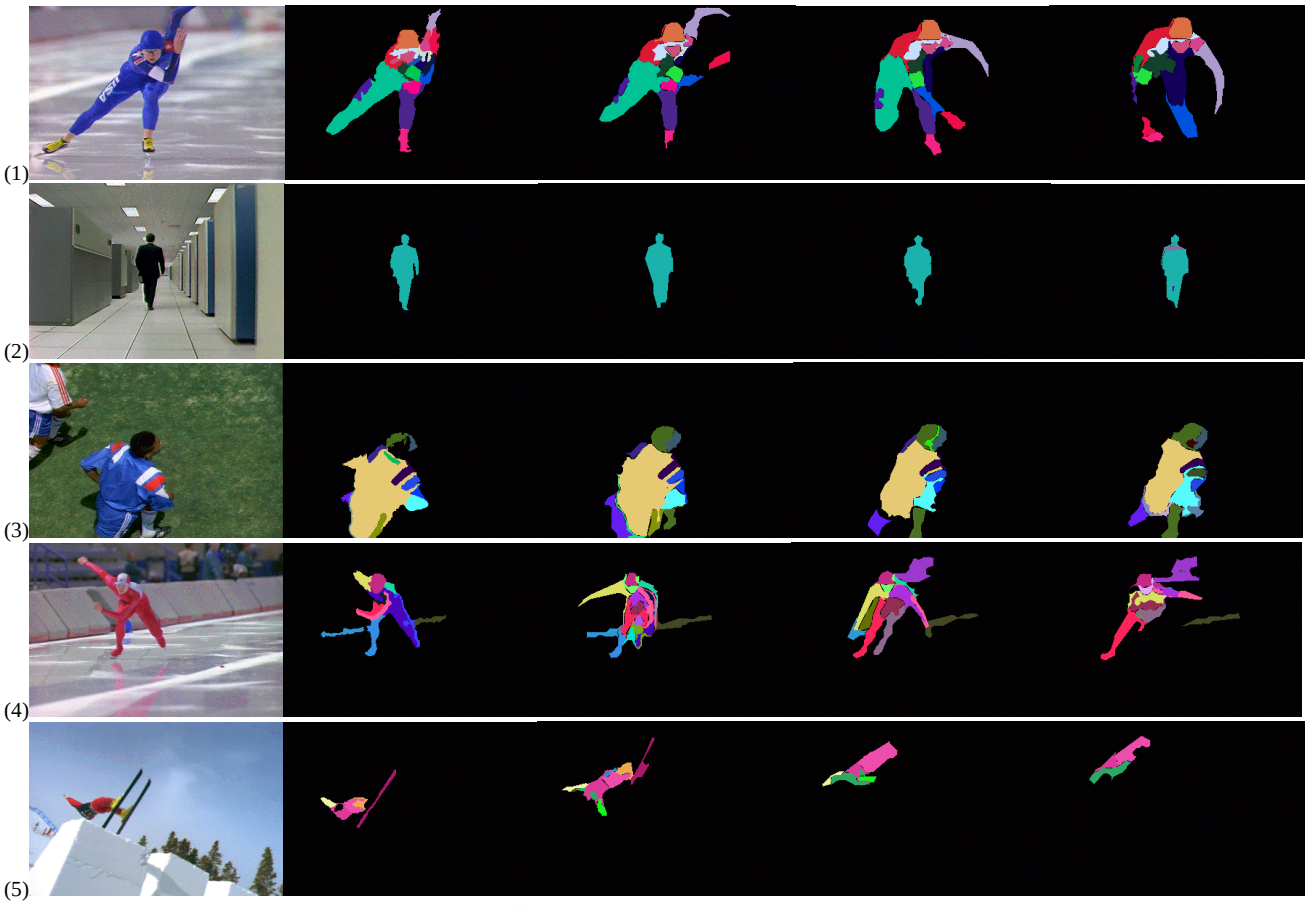

Figure 3. Moving object detection and tracking results of five image sequences (detected objects are show at frame #1, #10, #20 and #30), test videos are kindly provided by actions, sports, adventures Inc. and hot shots cool cuts Inc. for research.