Motion Segmentation in Long Image Sequences

Trupti Patil and Yujie Dong
Department of Electrical & Computer Engineering
Clemson University

Abstract

Analysis of long image sequence is important for visual surveillance, mobile robotics, and areas where a dynamic scene is observed over a long period of time, which means that a compact representation is needed to efficiently process it. In this report a novel representation for motion segmentation in long image sequences is presented. This representation, the feature interval graph, measures the pairwise rigidity of features in the scene. The feature interval graph is computed every several frames, making it a compact representation, uses an interval model of uncertainty and forms the basis for motion segmentation Results of this algorithm are presented on synthetic and real-world scenes.

1. Introduction:

In motion segmentation the goal is to cluster features extracted from a scene, into regions having a common motion. The clusters correspond to independently moving objects in the scene and so are useful for tracking and navigation.

Our project is based on the paper Motion Segmentation in Long Image Sequences, where authors Mills et al. [3] propose a compact representation of motion segmentation called Feature Interval Graph, which measures the pair-wise rigidity of features in a scene. This graph is then used to develop a new algorithm for motion segmentation.

Although we bypass occlusion analysis and appearance of new features in the scene, our contribution in this project has been implementing some methods that were not presented clearly in the original paper. For example, the paper just mentions that the cut vertex's classification is ambiguous, which we address by finding a more reasonable definition to achieve better segmentation results.

This report is organized as follows. We begin by presenting some previous work on long image segmentation and clustering. Then, we describe Mills and Novins' segmentation algorithm and the variant we considered. Finally, we present our results obtained from our variation of the original proposal

2. Previous Work:

A common approach in much of the previous work in long sequence analysis makes is to track features using a Kalman filter, and to group markers together if they have similar kinematic parameters - rotational velocity, translational velocity and acceleration.

Zhang and Faugeras [5] track 3D line features by extracting them from a scene using stereo and estimate their motion using an extended Kalman filter, grouping together features with similar motion. Due to the non-linear relationship between a feature's location and kinematics, the Kalman filter is extended by adding a linearization stage. To account for uncertainty, the Mahalanobis distance is used to compare features. If the distance between two features' parameters is small then they are determined to lie on the same rigid object.

Smith and Brady's [4] have a similar approach, ASSET-2, but uses 2D point features. This has the advantage that point features may be found in a wide variety of images and need only a monocular image sequence.

Although, the Kalman filter is efficient, since all of the computations are linear, is a recursive procedure, and is also robust with respect to measurement errors, the covariance matrices model the error assuming it to be Gaussian in distribution. For this assumption to be true, an a priori knowledge of the error model is required.

3. Method:

This section presents Mills and Novins' novel representation and algorithm for long sequence segmentation along with our variations to it. The representation called the Feature Interval Graph. The feature interval graph is computed every several frames, relying on only the current and previous frames, is efficient to compute and is robust with respect to measurement uncertainty.

3.1 Feature Interval Graph:

The model *Feature Interval Graph* is a graph $G_f = (V , E)$, where

- V is the set of vertices representing set of features identified in each frame.
- E is the set of edges, linking each pair of distinct features. Associated with each edge is a measurement of the two-dimensional distance between the features, computed using interval arithmetic.

The feature interval graph is initialized with the first observation and is constructed from two-dimensional features identified in each frame by using Harris feature detector [1] and tracked using Lucas-Kanade tracking algorithm [2]. Each subsequent frame of the sequence gives a new set of observations and measurements made for the distance between each pair of points. To combine the

information from multiple observations over the sequence, the distances are represented as intervals. The use of intervals allows a small amount of motion between features, and is considered the effect of measurement errors. The new distance stored in the graph is the intersection of the old distance and the latest observation. If this intersection is ever empty, it implies the change in distance between the two features must is too large to be accounted for by the estimated error bounds, and so the corresponding edge is removed from the graph. Over time, the graph evolves and compactly represents the information that has been over the sequence.

3.2 Segmenting Feature Interval Graph:

Segmentation of the feature interval graph is performed, to identify and separate objects from each other and the scene, through a triangle-based clustering. Here features are only assigned to the same cluster if they are both part of a triangle with a shared edge in the feature interval graph.

This segmentation yields a *Triangle Graph,* $G_t = (V , E)$, where

- V is the set of vertices corresponding to triangles in the feature interval graph.
- E is the set of edges, linking two distinct vertices if the corresponding triangles share an edge (or equivalently two vertices) in the feature interval graph.

Figure 1 below shows a graph which contains a number of problematic features. The feature interval graph (left) appears to contain three objects, indicated by the dashed lines. The segmentation is complicated by the missing edge, the spurious edge, and the cut vertex. To overcome the problems, we construct the corresponding triangle graph. Once the triangles in the graph have been labeled, the segmentation is transferred back to the vertices of the feature interval graph.

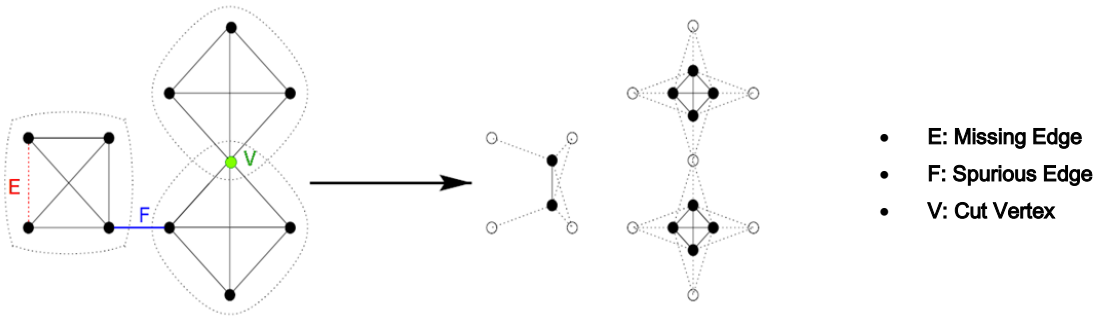

Figure 1: The triangle-based segmentation (right) has vertices (filled circles) constructed from connected triples of vertices in the feature interval graph (hollow circles). The components of the triangle graph correspond to desired segmentation.

In most cases, this is a straightforward process, however, a cut vertex contributes to triangles on two or more objects. Mills and Novins declare that classification of such vertices is ambiguous. However, we tackle this ambiguity in a unique manner. Figure 2 below helps illustrate our concept better.

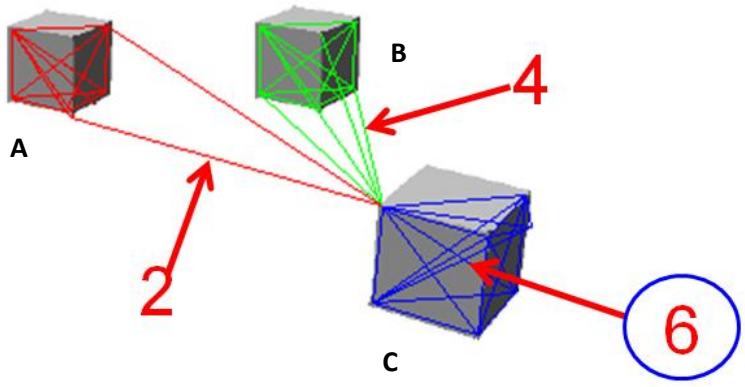

Figure 2: The ambiguous vertex has 2, 4 and 6 edges contributing to triangles on objects A, B and C

From the figure we see that the ambiguous vertex contributes most number of edges to triangles on object C. Hence it is assigned to that object. In case of equal number of contributing edges to triangles on all connected objects, the vertex is simply assigned randomly to any object. This method shows encouraging practical results as will be seen in the following section.

4. Results:

Figure 3(a) shows the interval graph of the first frame of a synthetic scene. We use interval value to represent the distances between different features. After several frames, if the distance between different features changes too much, we just remove the edge from the graph. Figure 3(b) shows the interval graph after 9 frames. However, we still see there are some edges between here. As a result, we create the corresponding triangle graph in Figure 3(c). Then we segment the image to get the final result as in Figure3(d).

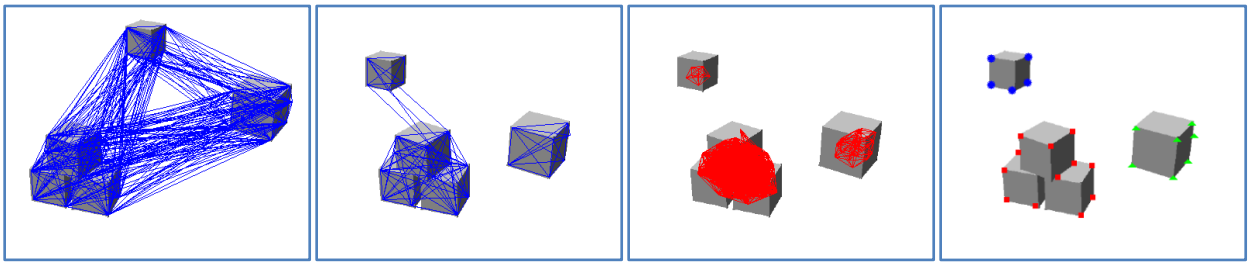

Figure 3: (a) (b) (c) (d)

Figure 4(a) shows the interval graph after 9 frames of another synthetic scene and Figure 4(b) is its corresponding triangle graph. Figure 4(c) is the result of segmentation. However, we notice a different colored vertex at the top. That is because in the interval graph, this point is a cut vertex; it can be labeled belonging to either object. Then we find that the vertex contributes most number of edges to the top object, hence, it is assigned to this object. Figure 4(d) shows the final result, which is more reasonable.

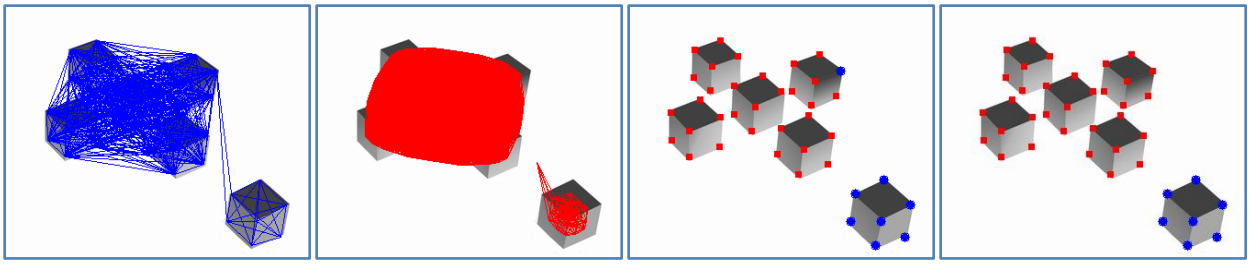

Figure 4: **(a)** **(b)** **(c)** **(d)**

Figure 5shows the result of motion segmentation of real image sequences. Only one car cannot be detected because the features tracked of this car drift too much, but other vehicles in the scene are tracked quite well.

Figure 5: Motion segmentation of real image sequence

5. Conclusion:

In this project, we implement the motion segmentation based on the paper Motion Segmentation in Long Image Sequences, where authors Mills et al. We use Interval Graph and Triangle Graph to segment different objects effectively. The feature interval graph differs from the Kalman filter in two main respects: an interval, rather than statistical, method of reasoning about uncertainty is used and no model of motion is needed. The Kalman filter uses a Gaussian model of uncertainty and a linear motion model. These both introduce parameters to the system which need to be known, or estimated. In addition, we

present a reasonable algorithm to label the cut vertex and the result is very promising. In the future, we will focus on the following work: First, we will deal with new objects appearing in the sequences; second, we will analyze the condition of occlusion.

6. References:

[1] C. Harris and M.J. Stephens. A combined corner and edge detector. *Alvey Vision Conference*, pages 147-152, 1988.

[2] B. D. Lucas and T. Kanade, An iterative image registration technique with an application to stereo vision. *Proceedings of Imaging understanding workshop*, pages 121—130, 1981.

[3] Steven Mills and Kevin Novins. Motion Segmentation in Long Image Sequences. *British Machine Vision Conference (BMVC).* pages 162-171, 2000.

[4] S. M. Smith and J. M. Brady. ASSET-2: Real-time motion segmentation and shape tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(9):814–820, 1995.

[5] Z. Zhang and O. D. Faugeras. Three-dimensional motion computation and object segmentation in a long sequence of stereo frames. *International Journal of ComputerVision*, 7(3):211–241, 1992.