SIMULTANEOUS OBJECT SEGMENTATION, MULTIPLE OBJECT TRACKING AND ALPHA MAP GENERATION

Yucel Altunbasak
Hewlett-Packard Laboratories
Broadband Information Systems Laboratories
1501 Page Mill Rd. MS 1U17
Palo Alto, CA, 94304

Remzi Oten and Rui J. P. de Figueiredo
Laboratory for Machine Intelligence, Vision and Neural Computing
University of California, Irvine, CA, 92697-2625

Abstract

This paper presents an object-based video modeling. Motion segmentation is performed at the initial frame to identify different coherently moving regions, called motion-objects. These regions are grouped to form objects. Each motion-object is fitted a content-based mesh, and tracked subsequently to the next frame via mesh motion estimation and compensation. The uncovered background (UB) region(s), which emerges when objects move, will be segmented so as to identify the new objects or occluded parts of already existing objects. The mesh model is modified to reflect the changes in object boundaries.

1 Introduction

Two-dimensional mesh-based motion estimation models impose a global connectivity constraint on the optic-flow field. The mesh connectivity is preserved by assigning a single motion vector to each node. All the patches connected to a particular node move it to the same location in the next frame. Although such a continuity is desirable within a single motion-object, it restricts the motion of the patches which are connected to the same node, but belong to differently moving objects. To allow motion discontinuity with the mesh models, but still to able to preserve the connectivity within a motion-object requires object-based modeling. Therefore, robust motion segmentation at the initial frame needs to be performed to achieve accurate object tracking.

[1]Yucel Altunbasak was with the Department of Electrical Engineering, University of Rochester when this work was initiated.

In this paper, we present a complete system in which the initial frame is segmented into differently moving regions. Each motion-object is fitted a content-based mesh, and tracked to the next frame. Object meshes will be modified at each frame to account for occlusions and any inaccuracy during motion estimation and mesh design stages.

This motion modeling leads to a more realistic description of video compared to the entire-frame-based mesh models discussed in [1]. Namely, the scene is segmented into motion-regions, and within each region, a deformable motion model is assumed. Mild deformations can be handled since each object is modeled as a set of connected triangles, and within each triangle the motion field is described by an affine transformation.

Section 2 describes motion segmentation and object-formation stage. In Section 3, object-scalable content-based mesh design will be described briefly. Then, Section 4 explains object-tracking via mesh-motion estimation. In Section 5 and 6, detection/segmentation of UB and refinement of object meshes will be discussed, respectively. In Section 7, a brief outline of the complete algorithm is given. In Section 8, example results will be provided on real video sequences. Finally, our conclusions and observations will be summarized in Section 9.

2 Object Segmentation

The object segmentation is only performed at the initial frame, and consists of two stages: i-) automatic motion segmentation ii-) semi-automatic region grouping. Motion segmentation refers to labeling pixels which are associated with different coherently moving parts/regions of the image. It is closely related to

0-8186-8183-7/97 $10.00

two other problems, motion detection and motion estimation. Motion detection, estimation, and segmentation are all plagued with two fundamental limitations: occlusion and aperture problems. For example, pixels in a flat image region may appear as if they are stationary even if they are moving due to the aperture problem; and/or erroneous motion vectors may be found for pixels in covered or uncovered image regions due to the occlusion problem. The occlusion and aperture problems are mainly responsible for misalignment of motion and actual object boundaries and over-segmentation of the motion field.

It is difficult to associate a generic figure of merit with a motion segmentation result. If motion segmentation is employed to improve the compression efficiency, then over-segmentation may not cause a concern. On the other hand, if it is used for object definition with application to object-based functionalities as in the upcoming MPEG-4 standard, then it is of utmost importance that resulting motion boundaries align with actual object boundaries. Although it may not be possible to achieve this perfectly in a fully automatic manner, elimination of outlier motion vector estimates and imposing spatio-temporal smoothness constraints on the segmentation map improve the chances of obtaining more meaningful segmentation results. We utilized the the algorithm proposed by one of the authors, and explained in paper [2], which attempts to achieve this latter goal without requiring extremely computationally demanding models and procedures. Then, we interactively group motion-objects to form objects.

3 Object-Based Mesh Design

The boundary of each object needs to be approximated by a shape model that can be represented with a few parameters. Most commonly employed shape models are polygonal and B-spline approximations. Here, we employ a polygonal approximation because of its simplicity and robustness. Furthermore, a polygonal boundary naturally coincides with the boundaries of the proposed mesh model.

There have been three approaches reported for content-based mesh design: optimization, split-and-merge, and spatio-temporal gradients methods. Designing an optimal mesh structure requires global optimization of a suitable cost function [3]. Split and merge methods successively divide, starting possibly with a uniform mesh, patches which do not satisfy a predetermined criterion in an attempt to find locally optimum solutions [4]. An efficient content-based mesh design method which utilizes spatio-temporal image intensity gradients was proposed by one of the authors [5].

Here, we utilized object-scalable content-based mesh. Object scalable mesh design is realized in three steps: i) Approximate the boundary of each individual object by a polygon, ii) utilize the node point selection algorithm described in [5] to place nodes inside *each object*, and ii) apply constrained Delaunay triangulation, where line segments representing the boundary of the object polygons are passed as constraints.

4 Object Motion Estimation

A node-point motion vector for each node needs to be estimated in order to perform motion compensation. If a node belongs to a particular motion-object, only the patches within that motion-object should be used for node-point motion estimation. More specifically, we would like to find the motion vector at the node N by, say, hexagonal matching [6] (See Fig. 1). Assuming that the node N belongs to object 1, triangles 1, 2, and 3 should be utilized instead of all seven triangles connected to it (labeled 1, 2, 3, 4, 5, 6 and 7 in Fig. 1). Similarly, if it is part of the object 2, then triangles 4, 5, 6, and 7 should be utilized instead. This allows the objects to move independent of each other when they are motion compensated. Hence, the mesh connectivity (therefore motion smoothness) along the motion-object boundaries is completely suppressed.

We utilized modified version of hexagonal matching to estimate node-point motion vectors [7]. To account for occlusions in motion estimation, we only utilized the pixels which do not fall into occlusion areas when they are motion-compensated.

5 Uncovered Background Segmentation

When the objects are compensated with their respective motion vectors, some regions in the reconstructed frame will be left uncompensated (See Fig. 1). These uncovered regions (UB) may include new objects entering into the scene, or parts of objects which are occluded in the previous frames. It is a challenging problem to determine which part of UB region belongs to which object? and which part of the UB region includes new objects?

We utilized the same object segmentation algorithm discussed in Section 2, with the difference that it is only applied to UB region rather than whole image. Namely, UB region is segmented into different motion regions, and these new motion-objects are assigned (merged) with one of the already existing motion-objects if they are found to be similar in terms of motion, color and texture characteristics. The parts of UB that are not merged with existing motion-objects

are labeled as new objects. Unfortunately, various problems occur: i-) UB is usually a small region, thus, it is difficult to perform a robust motion segmentation on it, ii-) It is also difficult to obtain motion, texture, color, and shape features of such a small region reliably, and iii-) There are cases in which similarity in terms of motion/color/texture does not necessarily mean that these regions should be merged. Therefore, there may be some mistakes after UB partitioning and merging stage. We interactively correct any mistake.

6 Mesh Refinement

Next, we need to modify the object-meshes considering the uncovered parts of the objects. The mesh tracked by the motion vectors may not well represent the new combined object. This problem could have been solved by the mesh refinement algorithm developed in [8]. However, sometimes, UB region(s) can be very long but thin shaped. Putting new nodes without considering already existing nodes may result in overpopulation of nodes along the boundaries. In such a case, we can perturb some of the already existing nodes such that perturbed mesh may well represent the combined object. Therefore we face the problem when to perturb the mesh? and when to redesign by putting new nodes?

We examine the shape of UB region by looking at its aspect ratio. If the aspect ratio is large, and either the width or the height of the UB region is smaller than a predefined threshold, then we perturb the mesh in order to make it aligned with the new boundary. Otherwise, we design a content-based mesh inside UB region, and merge with the tracked mesh.

7 Complete Algorithm

The brief outline of the complete algorithm can be given as follows:

1. Set frame index $k = 1$.
2. Estimate the dense motion field from the frame k to the frame $k+1$. Perform motion segmentation on frame k using the dense motion flow as explained in Section 2. Form objects as collections of motion-objects (See Section 2).
3. Approximate the boundary of each motion-object [1], and design a content-based mesh for each object as described in [9].
4. Compute motion estimates for each node of each object as explained in [7].
5. Motion-compensate each object by node-point motion vectors. Uncompensated points on the frame $k+1$ are labeled as UB regions. Partition UB as explained in Section 5. Assign each part of UB to one of already existing motion-objects or label it as a new motion-object. Compute the new shapes of each object, and store the updated object bit-maps as the alpha-map on frame $k+1$.
6. Refine the mesh model of each object as explained in Section 6.
7. Increase k by 1. Go to Step 2.

8 Results

The concepts of simultaneous object-segmentation and multiple object tracking are illustrated on the "Tennis" sequence.

We have performed an object segmentation as explained in Section 2 on the first frame of "Tennis" sequence. Figure 2 (a) depicts the object segmentation map. Four objects are extracted and shown with a different color. The content-based mesh design algorithm explained in [8] is applied on each object on the first frame. Figure 2 (b) depicts the meshes associated with "Table","Ball","Person" and "Background" objects overlayed on the image.

Each object is tracked by the mesh tracking algorithm described in Section 4. The location of each object after motion compensation is shown in Figure 2 (c), while the uncovered background (UB) region on the next frame is shown in Figure 2 (d). Figure 2 (e) and (f) show the modified (refined) mesh at the 2^{nd} and 20^{th} frames, respectively.

9 Conclusions

A complete object-oriented mesh-based video modeling is presented where motion segmentation is performed at the initial frame to identify different coherently moving regions, which are grouped to form objects. Each motion-object is fitted a content-based mesh, and tracked subsequently to the next frame via mesh motion estimation and compensation. Each object mesh model is refined to account for the occlusions and newly appearing/disappearing objects.

The proposed method can provide alpha-map to an MPEG-4 encoder as well as motion information. It can also be utilized as an "occlusion adaptive" dense motion estimation algorithm. Object tracking has also applications in "Special Effects Authoring".

The current research focuses on how to further optimize each stage in terms of speed and robustness. We also try to minimize the user-interactivity, and identify better forms of interaction with the computer.

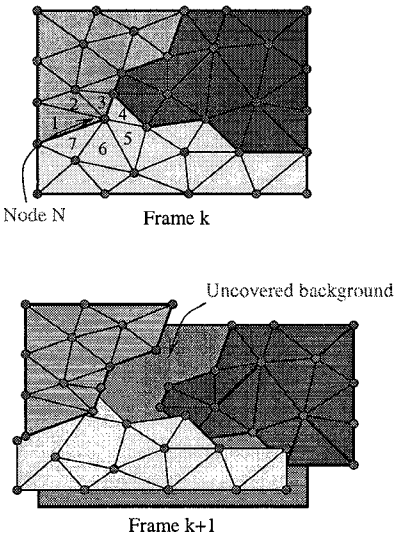

Figure 1: Demonstration of principles of multiple object tracking with 2-D meshes.

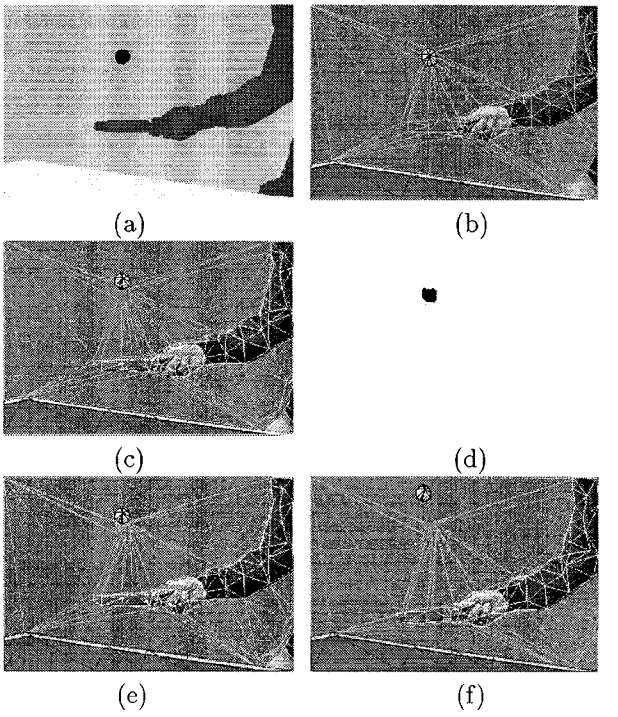

Figure 2: Illustration of proposed algorithm on "Tennis" sequence. a) Object segmentation map; b) Object-meshes are overlaid on the first frame of the "Tennis" sequence; c) Location of the objects on the 2^{nd} frame; d) Uncovered background regions e) and f) Modified object-meshes on the 2^{nd} and 20^{th} frames, respectively.

References

[1] Y. Altunbasak and A. M. Tekalp, "Occlusion-adaptive 2-d mesh tracking," in *Proc. IEEE Int. Conf. Accoust. Speech and Sig. Proc.*, (Atlanta, Georgia), May 1996.

[2] P. Eren, Y. Altunbasak, and A. M. Tekalp, "Region-based affine motion segmentation using color information," in *Proc. IEEE Int. Conf. Accoust. Speech and Sig. Proc.*, (Munich, Germany), 1997.

[3] A. Wan and D. Cai, "Fast automatic face feature ints extraction for model-based image coding," in *Picture Coding Symposium*, pp. 282–283, Sept. 1994.

[4] Y. Wang, R. S. Wang, O. Lee, T. Chen, H. H. Chen, and B. G. Haskell, "Mouth shape detection and tracking using an active mesh," *SPIE Visual Communications and Image Processing*, vol. 2501, pp. 1141–1152, May 1995.

[5] Y. Altunbasak and A. M. Tekalp, "Object-scalable content-based 2-d mesh design and tracking for object-based video coding," *IEEE Transactions on Image Processing*, September 1997.

[6] Y. Nakaya and H. Harashima, "Motion compensation based on spatial transformations," *IEEE Trans. Circuits and Syst. Video Tech.*, vol. 4, pp. 339–357, June 1994.

[7] Y. Altunbasak and A. M. Tekalp, "Very low bitrate video coding using object-based mesh design and tracking," in *SPIE/IS&T Symp. Electronic Imaging Sci. & Tech.*, (San Jose, California), 1996.

[8] Y. Altunbasak, A. M. Tekalp, and G. Bozdagi, "Two-dimensional object-based coding using a content-based mesh and affine motion parameterization," in *Proc. IEEE Int. Conf. Image Proc.*, (Washington D.C.), Oct. 1995.

[9] Y. Altunbasak and A. M. Tekalp, "Content-based mesh generation for very low bitrate video coding," in *Symposium on Multimedia Communications and Video Coding*, (New York City, NY), Oct. 1995.