2-D MESH-BASED SYNTHETIC TRANSFIGURATION OF AN OBJECT WITH OCCLUSION

Candemir Toklu *A. Tanju Erdem*[†] *A. Murat Tekalp*

Department of Electrical Engineering and Center for Electronic Imaging Systems
University of Rochester, Rochester, NY 14627
[†] Department of Electrical Engineering
Bilkent University, Ankara, Turkey 06533

ABSTRACT

This paper addresses 2-D mesh-based object tracking and mesh-based object mosaic construction for synthetic transfiguration of deformable video objects with deformable boundaries in the presence of another occluding object and/or self-occlusion. In particular, we update the 2-D triangular mesh model of a video object incrementally to account for the newly uncovered parts of the object as they are detected during the tracking process. Then, the minimum number of reference views (still images of a replacement object) needed to perform the synthetic transfiguration (object replacement and animation) is determined (depending on the complexity of the motion of the object-to-be-replaced), and the transfiguration of the replacement object is accomplished by 2-D mesh-based texture mapping in between these reference views. The proposed method is demonstrated by replacing an orange juice bottle by a cranbery juice bottle in a real video clip.

1. INTRODUCTION

Many multimedia applications, such as augmented reality, bitstream editing and interactive TV, demand object-based video modeling and manipulation. Synthetic object transfiguration refers to replacing an image object in a real video clip with a synthetic and/or natural object via digital post-processing. Successful transfiguration requires accurate tracking of the boundary, local motion and intensity (contrast and brightness) variations of the image object that is to be replaced.

Existing methods for object tracking can be broadly classified as boundary (and thus shape) tracking, [1, 2], and region tracking methods, [3, 4]. However, none of the above boundary or region tracking methods address tracking the local motion of the object. Local deformations within an region may be estimated by means of dense motion estimation or 2-D mesh-based representations. Prior work in mesh-based motion estimation and compensation includes [5, 6, 7]. These methods however, do not address tracking of an arbitrary object in the scene, since they treat the whole frame as the object of interest.

This paper addresses 2-D mesh-based tracking and transfiguration of deformable objects with deformable boundaries in the presence of another occluding object and/or self-occlusion. It extents the prior work by the authors [8, 9, 10]. In [8], the boundary of the object was modeled by a polygon with a small number of vertices, and the interior of the object was modeled by a uniform mesh under the "mild deformation" assumption which did not allow for occlusions or significant deformations of the object boundary. We attempt to release these restrictions in [9], where the boundary of the object is modeled by an active contour [1] to better track its deformations (e.g., an object undergoing out-of-plane rotation); and improved motion estimation and triangulation methods are employed to allow for possible occluding objects and/or self-occlusion (e.g., covered/uncovered regions). However, tracking deficiencies have been observed around occlusion boundaries of fast moving objects. In particular, if an occluding object splits the object-to-be-tracked into multiple pieces by partially covering it, the relationship between the multiple pieces is lost. In order to address this problem, we developed an improved mesh-based tracking method in [10], which constructs a mesh-mosaic of the object as the object is being tracked. This is accomplished by updating the 2-D triangular mesh to incrementally append the uncovered parts of the object as they are detected during the tracking process.

In this paper, we employ the mesh-based object mosaic constructed as described in [10] for the purpose of synthetically transfiguring a replacement object in place of the actual object being tracked. Once the tracking is performed and object mosaic is created, the minimum number of views of the replacement object needed for transfiguration is determined, and a replacement object mosaic is constructed from these views. The replacement object mosaic is then used for creating a new video clip where the replacement object goes through the same motion as the object to be replaced. Section 2 summarizes the algorithm for tracking and object mosaic construction. The details of the transfiguration of the replacement object are described in Section 3. We demonstrate the tracking and transfiguration performance of the proposed methods on a rotating bottle sequence in Section 4.

2. OBJECT TRACKING AND MOSAIC CONSTRUCTION

We assume that the user manually selects the contour enclosing the object to be tracked in the first frame of the im-

*This work was supported in part by Kodak, an IUCRC grant from National Science Foundation and a New York State Science and Technology Foundation grant

age sequence. This contour is snapped to the actual object boundary on the first frame of the image sequence using energy minimization [10]. A content-based adaptive mesh [11] is then fit inside the contour. This mesh is called the "reference mesh." Image region in the first frame covered by the reference mesh is taken as the initial "object mosaic." At each frame, the image region covered by the mesh on that frame is assumed to be the warped and/or occluded version of the object mosaic. The mosaicking technique proposed in [10] and reviewed below is more general than those given in [12] because it allows for (i) local motion as opposed to global transformations of the object and (ii) out-of-plane rotations of the object that result in self-occlusions. Given the reference mesh and the initial object mosaic, the following steps are carried out to track and construct the object mosaic:

1. *Find the object-to-be-covered (OTBC) regions in the current frame and the covered parts of the object mosaic:* Estimate forward dense motion field inside the object boundary in the current frame. Use forward dense motion field to compute the Displaced Frame Difference (DFD) within the object boundary and threshold to obtain the OTBC regions. Map these regions back to object mosaic using the affine transformations between the reference mesh and the current mesh to update the covered parts of the object mosaic.
2. *Predict the mesh in the next frame:* The nodes of the current mesh which do not fall inside OTBC regions in the current and/or previous frames are moved to the next frame by sampling the dense motion field. We refer to these nodes as *visible nodes*. The location of the all other nodes, which are called *occluded nodes*, in the next frame are predicted by looking to node location histories and fitting an affine motion trajectory model in the least square sense. The mesh reconstructed in the next frame is called the *predicted mesh*.
3. *Find uncovered object (UO) regions in the next frame:* All uncovered parts of the object mosaic are warped into next frame to estimate the boundary of the object in the next frame. This boundary is then snapped to nearby image edges. The image regions inside the snapped boundary but outside the predicted boundary are identified as the UO regions.
4. *Update the reference mesh and the predicted mesh on the next frame:* Given the UO regions, the visible boundary nodes of the predicted mesh are pulled onto object boundary such that their new location minimizes a predefined cost function. Every visible boundary node that fall outside the object boundary is pulled to the nearest point on the boundary. On the other hand, every visible boundary node that fall inside the object boundary is pulled to a point on the boundary such that (i) the node is close to its previous position, and (ii) the distance between the points which are obtained by mapping the node into the object mosaic by the affine mappings of the triangles where the node is a vertex of the triangles is minimum. Use one of the these affine mappings to predict the new location of the boundary node in the object mosaic. If the area of a patch with at least one vertex being a boundary node gets larger than a predefined threshold, then it is divided into smaller patches.
5. *Update the object mosaic:* Use the constrained Hexagonal Search discussed in [8], to refine the boundary nodes of the reference mesh and the inside nodes of the predicted mesh to form an *updated mesh* such that the intensity prediction error within the object boundary in the next frame is minimized. Given the refined reference mesh and the updated mesh the intensity distribution within UO regions are warped into the object mosaic.

3. SYNTHETIC TRANSFIGURATION

Given the object mosaic and the reference and updated meshes for every frame in the image sequence, we construct a set for every pixel in the object mosaic from the indices of the frames where the pixel is visible. Let M and N denote the number of pixels in the object mosaic and the number of frames in the image sequence, respectively. Also let $\mathcal{S}_m$ denote the index set obtained for the mth pixel in the object mosaic. Initially, we label all the pixels in the object mosaic as *unmarked*. We then pick the first frame to be the first view and relabel all pixels m in the object mosaic as *used* if $1 \in \mathcal{S}_m$. The following steps are carried out to relabel the remaining *unmarked* pixels in the object mosaic and hence to determine the views to be used for transfiguration:

1. Obtain a sequence $a_n,\ n = 1, \cdots, N$ of numbers, where a_n denotes the number of *unmarked* pixels in the object mosaic that come from the nth frame.
2. Find the maximum of $a_n,\ n = 1, \cdots, N$, and let a_p denote this maximum.
3. If a_p is greater than a predefined threshold then select pth frame as the next view and relabel all pixels m in the object mosaic as *used* if $p \in \mathcal{S}_m$, and go to Step 1. Otherwise stop.

The user is assumed to have the still images of the replacement object at the views obtained above. We further assume that a global spatial transformation between the replacement object and the object to be replaced in each view can be found. Using these transformations, the updated mesh in each view is mapped onto the replacement object in the corresponding view. Then, the views of the replacement object are warped into the object mosaic to create the replacement object mosaic. The intensity value of every *unmarked* pixel on the object mosaic is spatially interpolated from the neighboring *used* pixels. Finally, the replacement object mosaic is used for rendering the motion of the object to be replaced in every frame of the given image sequence to achieve its transfiguration.

4. RESULTS

We demonstrate the performance of the proposed approach in the case of self occlusion due to out-of-plane rotation of the object-to-be-tracked. The test sequence is called "Rotating Orange Juice Bottle" and is recorded by a rigid Hi-8 mm comsumer camcorder and contains a rotating object in

front of a stationary background. We held the camera stationary and let the bottle rotate. Interlace-to-progressive conversion of the sequence is done by spatially interpolating the even fields to frame resolution (300 lines by 330 pixels). The sequence is very noisy and due to the transparent nature of the object and the background colors, the information on the bottle to be tracked is low in contrast. Therefore we have provided the object/background segmentation in each frame of the video clip as an input to our tracking/mosaicking algorithm. The proposed algorithm is tested on the first 10 frames of the sequence. The original frames $1, 4, 7$, and 10 of the sequence are provided in raster scan order in Fig. 1. In Fig. 2, we show the tracked meshes overlaid on the same frames $1, 4, 7$, and 10 in raster scan order. The algorithm decribed in Section 3 selected the 1st and the 10th frames of the sequence as the views to reconstruct the whole sequence which are given in Fig. 1. The static mosaic object created after the 10th frame of the sequence and reconstructed frames $4, 7$, and 10 are displayed in Fig. 3. Since we find the index sets for every pixel in the object mosaic we know which pixels on the object mosaic are visible in frames 1 and/or 10. We use this information to enhance the quality of the texture mapping and map the intensities of frames 1 or 10 onto each object in the sequence to obtain the intensity of the object.

We then shoot a video clip of the cranberry juice bottle with the same camcorder from a similar perspective as the bottle is manually rotated in a similar way as the orange juice bottle. We identify a frame in this sequence that matches the first frame of the orange juice bottle sequence in perspective and outline the boundary of the bottle. Then, perpective transformation parameters are calculated between the boundaries of the two bottles in those two corresponding frames. The texture of the cranberry juice bottle object is mapped onto the orange juice bottle object in frame 1 using the computed transformation parameters. The same steps are carried out for the frame 10 of the orange juice bottle sequence and its matching frame in the cranberry sequence to obtain the corresponding texture map for the frame 10 of the orange juice bottle replacing it with the cranberry juice bottle. Using the two corresponding texture maps for frames 1 and 10 of the orange juice bottle sequence, a rotating cranberry juice bottle with the same motion as the orange juice bottle is reconstructed. In Fig. 4 we show the frames $1, 4, 7$, and 10 of the transfiguration sequence in raster scan order.

5. CONCLUSIONS

It is clear that the visual quality of the synthetically transfigured video objects strongly depends on the accuracy of tracking of the actual video object to be replaced. The number of views (still images) of the replacement object needed for transfiguration depends on the complexity of the motion of the video object-to-be-replaced. When there are newly uncovered regions within the frames of interest, due to the motion of the video object (e.g., out-of-plane rotations), we need multiple views of the replacement object which shows all the texture of the replacement object needed to perform the transfiguration. The algorithm presented in this paper determines and employs the minimum number of views necessary for transfiguration. Note also that, if the tracking is lost due to the complexity of the object motion, a view of the replacement object similar to that of the object-to-be-replaced at the frame where tracking is lost would be needed to reinitialize the process. In the example shown above, this was not needed as the tracking of the video object was satisfactory.

6. ACKNOWLEDGMENTS

We would like to thank to Dr. J. Riek and Dr. S. Fogel of Eastman Kodak Company and Dr. P. J. L. van Beek for their contributions to the software used in this work.

REFERENCES

[1] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: active contour models. *Int. Journal of Comp. Vision*, 1(4):321–331, 1988.

[2] C. Kervrann and F. Heitz. Robust tracking of stochastic deformable models in long image sequences. In *IEEE Int. Conf. Image Proc.*, Austin, TX, November 1994.

[3] Y. Y. Tang and C. Y. Suen. New algorithms for fixed and elastic geometric transformation models. *IP*, 3(4):355–366, July 1994.

[4] F. G. Meyer and P. Bouthemy. Region-based tracking using affine motion models in long image sequences. *CVGIP: Image Understanding*, 60(2):119–140, Sept. 1994.

[5] Y. Nakaya and H. Harashima. Motion compensation based on spatial transformations. *IEEE Trans. Circuits and Syst. Video Tech.*, 4(3):339–357, June 1994.

[6] Y. Wang and O. Lee. Active mesh-a feature seeking and tracking image sequence representation scheme. *IEEE Trans. Image Processing*, 3(5):610–624, Sept. 1994.

[7] R. Szeliski and H.-Y. Shum. Motion estimation with quadtree splines. Technical report, 95/1, Digital Equipment Corp., Cambridge Research Lab, Mar. 1995.

[8] C. Toklu, A. T. Erdem, M. I. Sezan, and A. M. Tekalp. Tracking motion and intensity variations using hierarchical 2-D mesh modeling,. In *GMIP*, volume 58, Number 6, pages 553–573, November 1996.

[9] C. Toklu, A. M. Tekalp, A. T. Erdem, and M. I. Sezan. 2-D mesh-based tracking of deformable objects with occlusion,. In *IEEE Int. Conf. Image Proc.*, Lausanne, Switzerland, Sept. 16-19 1996.

[10] C. Toklu, A. M. Tekalp, and A. T. Erdem, 2-D triangular mesh-based mosaicking for object tracking in the presence of occlusion,. In Proceedings of SPIE Visual Communications and Image Processing Conference, Volume 3024, San Jose, CA, Feb. 8-14, 1997.

[11] 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.

[12] M. Irani, P. Anandan, and S. Hsu. Mosaic based representation of video sequences and their applications. In *Int. Conf. Computer Vision*, pages 605–611, Cambridge, MA, June 1995.

Figure 1.

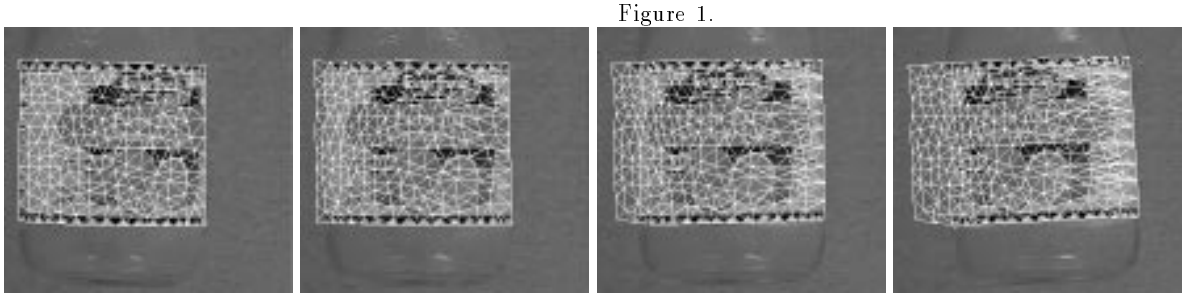

Figure 2.

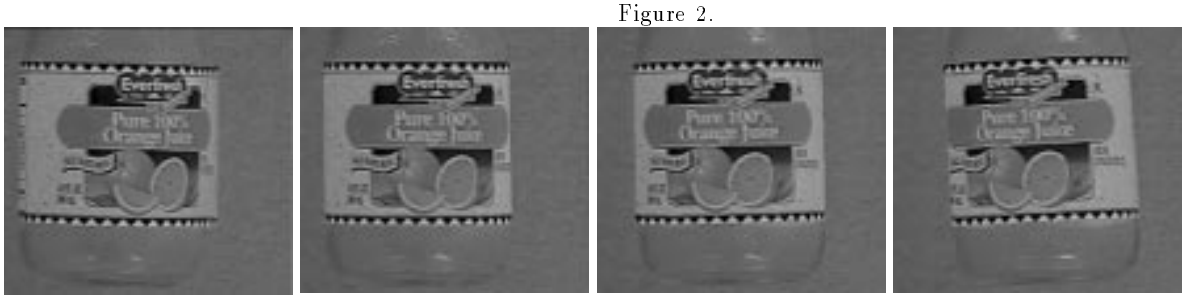

Figure 3.

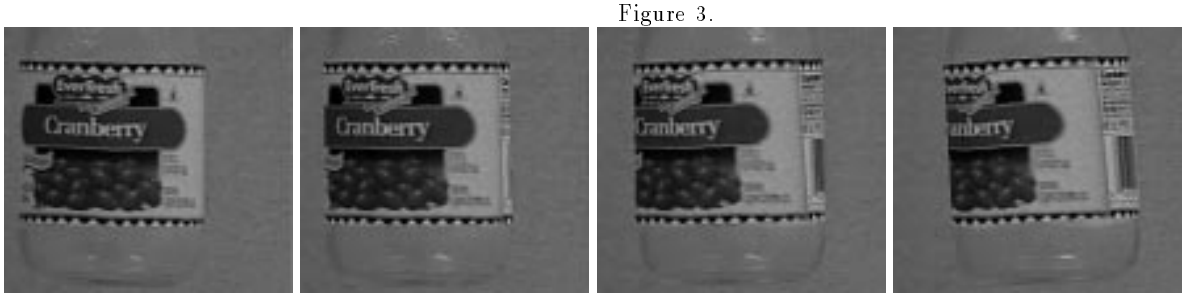

Figure 4.