Camera-Based In-situ 3D Modeling Techniques for AR Diorama in Ubiquitous Virtual Reality

Atsushi Umakatsu[1], Hiroyuki Yasuhara[1],
Tomohiro Mashita[1,2], Kiyoshi Kiyokawa[1,2], and Haruo Takemura[1,2]

[1] Graduate School of Information Science and Technology, Osaka University,
1-5 Yamadaoka, Suita, Osaka 565-0871, Japan
[2] Cybermedia Center, Osaka University,
1-32 Machikaneyama, Toyonaka, Osaka 560-0043, Japan
{mashita,kiyo,takemura}@ime.cmc.osaka-u.ac.jp

Abstract. We have been studying an in-situ 3D modeling and authoring system, AR Diorama. In the AR Diorama system, a user is able to reconstruct a 3D model of a real object of concern and describe behaviors of the model by stroke input. In this article, we will introduce two ongoing studies on interactive 3D reconstruction techniques. First technique is feature-based. Natural feature points are first extracted and tracked. A convex hull is then obtained from the feature points based on Delaunay tetrahedralisation. The polygon mesh is carved to approximate the target object based on a feature-point visibility test. Second technique is region-based. Foreground and background color distribution models are first estimated to extract an object region. Then a 3D model of the target object is reconstructed by silhouette carving. Experimental results show that the two techniques can reconstruct a better 3D model interactively compared with our previous system.

Keywords: AR authoring, AR Diorama, 3D reconstruction.

1 Introduction

We have been studying an in-situ 3D modeling and authoring system, AR Diorama [1]. In the AR Diorama system, a user is able to reconstruct a 3D model of a real object of concern and describe behaviors of the model by stroke input. Being able to combine real, virtual and virtualized objects, AR Diorama has a variety of applications including city planning, disaster planning, interior design, and entertainment. We target smart phones and tablet computers with a touch screen and a camera as a platform of a future AR Diorama system.

Most augmented reality (AR) systems to date can play only AR contents that have been prepared in advance of usage. Some AR systems provide in-situ authoring functionalities [2]. However, it is still difficult to handle real objects as part of AR contents on demand. For our purpose, online in-situ 3D reconstruction is necessary. There exist a variety of hardware devices for acquiring a 3D model of a real object in a short time such as real-time 2D rangefinders. However, a special

R. Shumaker (Ed.): Virtual and Mixed Reality, Part I, HCII 2011, LNCS 6773, pp. 113–122, 2011.

hardware device is not desired in our scenario. In addition, acquiring geometry of an entire scene is not enough. In AR Diorama, we would like to reconstruct only an object of interest. Introducing a minimal human intervention is a reasonable approach for this segmentation problem since AR Diorama inherently involves computer human interaction. On the other hand, single camera-based interactive 3D reconstruction techniques have been intensively studied recently in the literatures of augmented reality (AR), mixed reality (MR) and ubiquitous virtual reality (UVR) [3,4,5,6]. These techniques only require a single standard camera to extract a target model geometry.

2 AR Diorama

Figure 1 shows an overview of the AR Diorama system architecture [1]. In the following, its user interaction techniques and a 3D reconstruction algorighm are briefly explained.

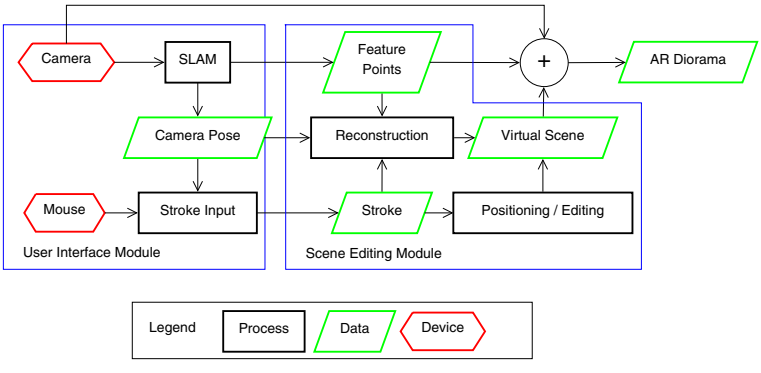


Fig. 1. AR Diorama system architecture [1]

2.1 User Interaction

AR Diorama supports a few simple stroke input-based interaction techniques to reconstruct and edit the virtual scene. First, a user needs to specify a *stage*, on which all virtual objects are placed, by circling the area of interest on screen. Then a stage is automatically created based on 3D positions of feature points in the area. Then the user is able to reconstruct a real object by again simply circling it. The polygon mesh of the reconstructed model is composed of feature points in the circle and the input image as texture. The reconstruction algorithm is described in more detail below. The reconstructed object is overlaid onto the original real object for later interaction. As the polygon mesh has only surfaces that are visible in the input image, the user will need to see the object from different angles and circle it again to acquire a more complete model. The reconstructed model can be saved to a file and loaded for reuse.

Fig. 2. Stroke-based scene editing in AR Diorama [1]

Once the model creation is done, the user is able to translate, rotate and duplicate the model. Translation is performed by simply draw a path from a model to the destination. Rotation is performed by drawing an arc whose center is on a model. Figure 2 shows an example of translation operation.

2.2 Texture-Based Reconstruction

In the AR Diorama system, a texture-based reconstruction approach has been used [1]. Natural feature points in the object region are first extracted and tracked using the open-source PTAM (parallel tracking and mapping) library [7]. A polygon mesh is created from 3D positions of the feature points by using 2D Delaunay triangulation calculated from the corresponding camera position. When the next polygon mesh is created from a different viewpoint, they are merged into a single polygon mesh. At this time, a texture-based surface visibility test is conducted and a false surface will be removed. That is, if a similarity between an apperance of a surface in a new mesh from the corresponding viewpoint and its corresponding appearance of a surface in the current mesh rendered from the same viewpoint using a transformation matrix between the two viewpoints is lower than a threshold, that surface is considered false and removed.

This is an easy-to-implement approach, however, the reconstruction accuracy is not satisfactory at all, mainly due to the limitation of 2D Delaunay triangulation. Examples of reconstructed models can be found in the middle column of Figure 3. To improve the model accuracy, we have implemented two different in-situ 3D reconstruction techniques inspired by recent related work, which we will report in next sections.

3 Feature-Based Reconstruction

First 3D reconstruction technique we have newly implemented is a feature-based one inspired by ProFORMA proposed by Pan et al [4]. In the following, its implementation details and some reconstruction results are described in order.

3.1 Implementation

Natural feature points in the scene are first extracted and tracked using the open-source PTAM (parallel tracking and mapping) library [7]. A PTAM's internal variable *outlier_count* is used to exclude unreliable feature points.

A Delaunay tetrahedralisation of the feature points is obtained using CGAL library. At this stage, a surface mesh of the obtained polygon is a convex hull of the feature points, and false tetrahedrons that do not exist in the target object need to be removed. While tracking, each triangle surface is examined and its corresponding tetrahedron will be removed if any feature point that should be behind the surface is visible. This carving process is expressed by the following equations.

$$P_{exist}(T_i|v) = \prod_v (T_i|R_{j,k}) = \prod_v (1 - Intersect(T_i, R_{j,k})) \tag{1}$$

$$Intersect(T_i, R_{j,k}) = \begin{cases} 1(\text{if } R_{j,k} \text{ intersects } T_i) \\ 0(\text{otherwise}) \end{cases} \tag{2}$$

T_i denotes the ith triangle in the model, j denotes a keyframe id for reconstruction, k denotes a feature point id, $R_{j,k}$ denotes a ray in the jth keyframe from a camera position to the kth feature point, v denotes all combinations of (j, k) where the kth feature point is visible in the jth keyframe. However, this test will remove tetrahedrons wrongly that exist in the target object due to some noise in feature point positions. To cope with this problem, we have implemented a probabilistic carving algorithm found in ProFORMA [4].

After carving, texture is mapped using keyframes, that are stored automatically during tracking, onto the polygon surfaces. A keyframe is added when the camera pose is different from any other camera poses associated with existing keyframes. As the camera moves around the object, a textured polygon model that approximates the target object is acquired.

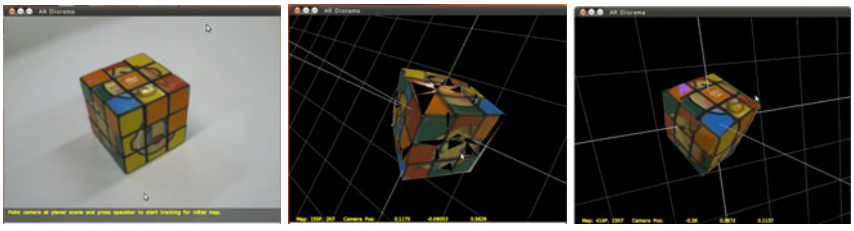
In ProFORMA, feature points on the target object are easily identified because the camera is fixated. In our system, a user can move a camera freely so segmentation of the target region from the background is not trivial. As a solution, the user roughly draws round an object of concern on screen in the beginning of model creation, to specify a region to reconstruct.

3.2 Results

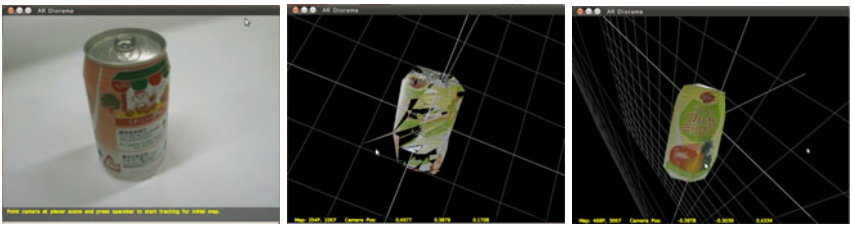
Two convex objects (a Rubic's Cube and an aluminum can) and a concave object (an L-shape snack box) were reconstructed by the implemented feature based reconstruction technique, and compared against those reconstructed by the previous technique [1]. A desktop PC (AMD Athlon 64 X2 Dual Core 3800+, 4GB RAM) and a handheld camera (Point Grey Research, Flea3, 648×488@60fps) were used in the system.

Figure 3 shows the results for a Rubic's Cube and an aluminum can. Virtual models reconstructed by the previous technique have many cracks and texture dicontinuities compared with the new technique.

Figure 4 shows the results for an L-shape snack box. Reconstructed object's shape approximates that of the target object better after carving, however, some tetrahedrons wrongly remain probably due to insufficient parameter tuning for the probabilistic carving. Another conceivable reason is tracking accuracy. In our

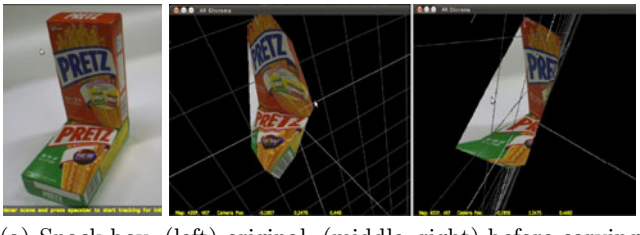


(a) Rubic's Cube. (left) original, (middle) old technique, (right) new technique

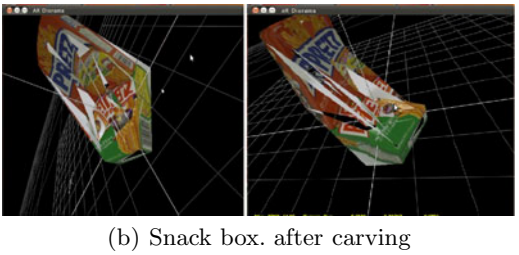


(b) Aluminum can. (left) original, (middle) old technique, (right) new technique

Fig. 3. Results for convex objects



(a) Snack box. (left) original, (middle, right) before carving



(b) Snack box. after carving

Fig. 4. Results for a concave object

system, position accuracy of feature points rely on the PTAM library whereas a dedicated, robust drift-free tracking method is used in ProFORMA.

4 Region-Based Reconstruction

A feature-based approach relies on texture on the object surface, and thus not appropriate for texture-less and/or curved objects. Second technique is a

silhouette-based approach inspired by an interactive modeling method proposed by Bastian et al [5]. In the following, its implementation details and some reconstruction results are described in order.

4.1 Implementation

Natural feature points in the scene are first extracted and tracked again using the PTAM library. Then a user draws a stroke on screen to specify a target object to reconstruct. The stroke is used to build a set of foreground and background color distributions in the form of a Gaussian Mixture Model, and the image is segmented into the two types of pixels using graph-cuts [8](initial segmentation). After initial segmentation, the target object region is automatically extracted and tracked (dynamic segmentation) using again graph-cuts. In dynamic segmentation, a distance field converted from a binarized foreground image in the previous frame is used for robust estimation. In addition, stroke input based interaction techniques, Inclusion brush and Exclusion brush, are provided to manually correct the silhouette.

After a silhouette of the target object is extracted, a 3D model approximating the target object is progressively reconstructed by silhouette carving. In silhouette carving, a voxel space is initially set around the object, and the 3D volume approximating the target object is iteratively carved by testing the projection of each voxel against the silhouette. This process is expressed by the following equation. v_t^i denotes the ith voxel in frame t ($v_0^i = 1.0$), $\boldsymbol{P}_t$ denotes a projection matrix, $W(\cdot)$ denotes a transformation from the world coordinate to the camera coordinate.

$$v_t^i = v_{t-1}^i f(I_t^\alpha(\boldsymbol{P}_t W(v^i))) \tag{3}$$

Normally a voxel will remain empty once removed. To cope with unstable PTAM's camera pose estimation, a voting scheme is introduced. That is, votes more than a threshold are required to finally remove a voxel. From the remaining voxel set, a polygon mesh is created using a Marching Cubes algorithm. Then each surface of the polygon mesh is textured based on the smallest angle between a camera pose in each keyframe and the surface normal.

4.2 Results

Using the same hardware devices as the feature-based reconstruction, it takes about 2.5 seconds from image capturing to rendering the updated textured object. However, the rendering and interaction performance is kept around 10 frames per second, thanks to a CPU-based, yet multi-threaded implementation. In the following, results of main steps of reconstruction as well as a few final reconstructed models are shown.

Figure 5 shows a segmentation result in a frame, a binarized image of the target object region, and the corresponding distance field. Foreground probability decreases rapidly near the silhouette. Figure 6 and Figure 7 show an example

(a) Segmentation result in the previous frame (b) Binarized image (c) Distance field

Fig. 5. Probability distribution in dynamic segmentation

(a) before (b) Inclusion brush in use (c) after

Fig. 6. Inclusion brush

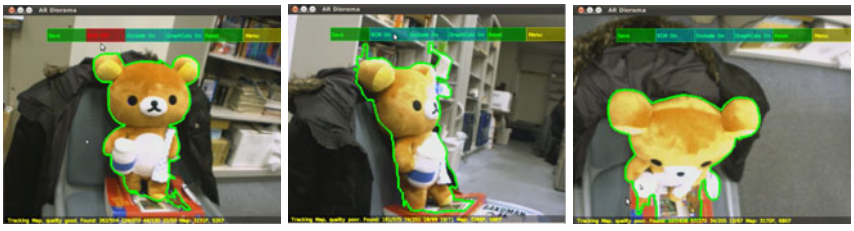
(a) before (b) Exclusion brush in use (c) after

Fig. 7. Exclusion brush

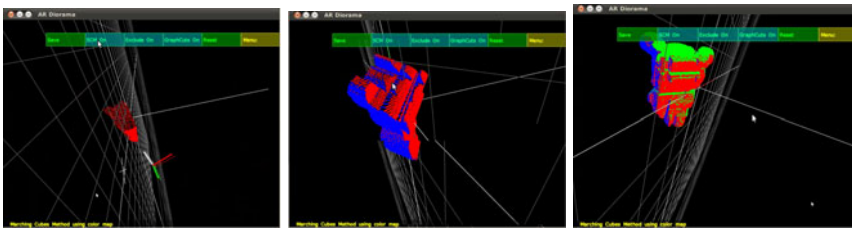
usage of Inclusion and Exclusion brushes, respectively. A user is able to add (remove) an area to (from) the foreground region interactively. Figure 8 shows a series of voxel data generated by silhouette carving in order of time (left to right). As the number of keyframes increases, the volume shape is refined to approximate the targe object. Voxel color indicates texture id.

Figure 9 shows a reconstructed plushie (c) and some keyframes used (a, b). Textures are mapped onto the model correctly, though some discontinuities appear. This is mainly due to brightness differences in textures mapped onto adjacent surfaces.

Figure 10 shows a reconstructed paper palace (c) and some keyframes used (a, b). In this case, a concave part is not reconstructed well as indicated in a green circle in Figure 10(c). This is a typical limitation of a simple silhouette carving

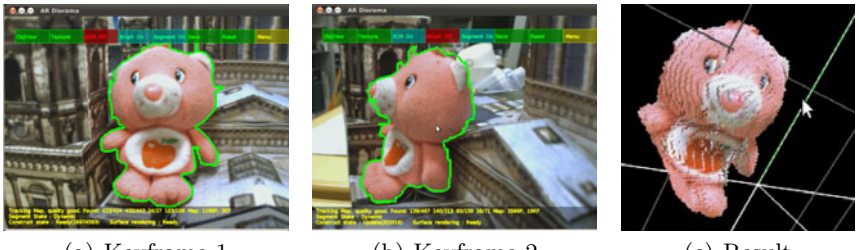


(a) Keyframes



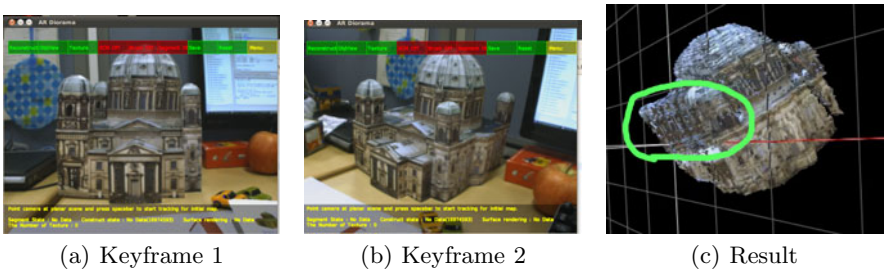
(b) Voxel data (color indicates texture id)

Fig. 8. Silhouette carving



(a) Keyframe 1 (b) Keyframe 2 (c) Result

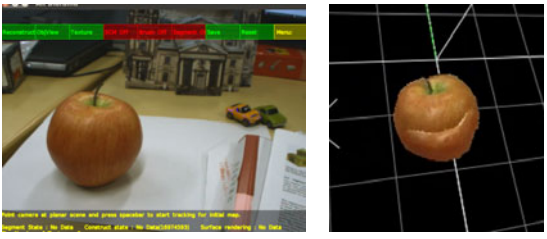
Fig. 9. Reconstruction of a plushie



(a) Keyframe 1 (b) Keyframe 2 (c) Result

Fig. 10. Reconstruction of a paper palace

algorithm. To tackle this, we will need to introduce photometric constraints or combine with a feature-based approach.



(a) Keyframe 1 (b) Result

Fig. 11. Reconstruction of an apple

Figure 11 shows a reconstructed apple (b) and a keyframe used (a). In this case no feature points were found in the foreground region and both our previous and new feature based techniques did not work. A region based technique is suitable for reconstructing such a feature-less object as far as its color distribution is different from that of the background.

5 Conclusion

In this study, we have introduced two types of implementations and results of 3D reconstruction techniques for our AR Diorama system, inspired by the recent advancements in this field [4,5].

The feature-based technique implemented has been proven to produce a better reconstructed model compared with our previous technique. Advantages of feature-based approaches over region-based approaches include that they can reconstruct concave objects, objects whose color distribution is similar to that of the background, and potentially non-rigid objects, and that users need not to shoot an object from many directions. However, with our current implementation, it was found that some nonexistent surfaces sometimes remain probably due to insufficient parameter tuning and inaccurate feature tracking.

The region-based technique has also been proven to produce a better reconstructed model compared with our previous technique. Advantages of region-based approaches include that they can reconstruct feature-less objects such as plastic toys and fruits. As far as the color distribution of the target object is different from that of the background, region-based approaches will succeed in reconstruction. However, they cannot handle concave objects well by its nature.

In the future, we will continue pursuing improving the reconstruction quality by combining feature-based and region-based approaches, extend the stroke-based interaction techniques [9,10,11,12] and develop an easy-to-use, multi-purpose AR Diorama system.

References

1. Tateishi, T., Mashita, T., Kiyokawa, K., Takemura, H.: A 3D Reconstruction System using a Single Camera and Pen-Input for AR Content Authoring. In: Proc. of Human Interface Symposium 2009, vol. 0173 (2009) (in Japanese)

2. Lee, G.A., Nelles, C., Billinghurst, M., Kim, G.J.: Immersive Authoring of Tangible Augmented Reality Applications. In: Proc. of the 3rd IEEE International Symposium on Mixed and Augmented Reality, pp. 172–181 (2004)
3. Fudono, K., Sato, T., Yokoya, N.: Interactive 3-D Modeling System with Capturing Support Interface Using a Hand-held Video Camera. Transaction of the Virtual Reality Society of Japan 10(4), 599–608 (2005) (in Japanese)
4. Pan, Q., Reitmayr, G., Drummond, T.: ProFORMA: Probabilistic Feature-based On-line Rapid Model Acquisition. In: Proc. of the 20th British Machine Vision Conference (2009)
5. Bastian, J., Ward, B., Hill, R., Hengel, A., Dick, A.: Interactive Modelling for AR Applications. In: Proc. of the 9th IEEE and ACM International Symposium on Mixed and Augmented Reality, pp. 199–205 (2010)
6. Hengel, A., Dick, A., Thormählen, T., Ward, B., Torr, P.H.S.: VideoTrace: Rapid Interactive Scene Modelling from Video. ACM Transactions on Graphics 26(3), Article 86 (2007)
7. Klein, G., Murray, D.: Parallel Tracking and Mapping for Small AR Workspaces. In: Proc. of the 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, pp. 1–10 (2007)
8. Boykov, Y., Kolmogorov, V.: An Experimental Comparison of Min-cut/max-flow Algorithms for Energy Minimization in Vision. IEEE Transactions on Pattern Analysis and Machine Intelligence 26(9), 1124–1137 (2004)
9. Thorne, M., Burke, D., van de Panne, M.: Motion Doodles: An Interface for Sketching Character Motion. ACM Transactions on Graphics 23(3), 424–431 (2004)
10. Cohen, J.M., Hughes, J.F., Zeleznik, R.C.: Harold: a world made of drawings. In: Proc. of the 1st International Symposium on Non-photorealistic Animation and Rendering, pp. 83–90 (2000)
11. Bergig, O., Hagbi, N., El-Sana, J., Billinghurst, M.: In-place 3D Sketching for Authoring and Augmenting Mechanical Systems. In: Proc. of the 8th IEEE International Symposium on Mixed and Augmented Reality, pp. 87–94 (2009)
12. Popovic, J., Seitz, S.M., Erdmann, M., Popovic, Z., Witkin, A.: Interactive Manipulation of Rigid Body Simulations. In: Proc. of the 27th Annual Conference on Computer Graphics and Interactive Techniques, pp. 209–217 (2000)