Sharing 3D Object with Multiple Clients via Networks Using Vision-Based 3D Object Tracking

Yukiko Shinozuka
Keio University
3-14-1 Hiyoshi Kohoku-ku
Yokohama, Japan
+81 - 45 - 566 - 1454
shinozuka@hvrl.ics.keio.ac.jp

Hideo Saito
Keio University
3-14-1 Hiyoshi Kohoku-ku
Yokohama, Japan
+81 - 45 - 566 - 1454
saito@hvrl.ics.keio.ac.jp

ABSTRACT

This paper proposes a new system for 3D object sharing with multiple clients using networks. Our system aims to support group communication across devices over the same 3D object and puts the annotations with augmented reality automatically. In order to keep putting the annotations during the interaction, we apply the vision-based 3D object tracking algorithm which estimates 3D position and pose of the object just by capturing the object with a single RGB camera. By employing the tracking algorithm robust to viewpoint changes and occlusion, we develop the 3D object sharing system with a desktop as a server and a laptop and a smart phone as clients. We conducted the experiments to show the tracking algorithm is tolerant of interaction and can be applied to various objects with texture.

Keywords

Augmented Reality, Networks, 3D Object Tracking

1. INTRODUCTION

It is popular to share video contents via network with multiple users for group communication. A user enjoys creating and sharing a video in Youtube [1] and Skype [2]. The user-driven content is called Consumer Generated Media (CGM), and the number of videos is increasing. However it is still difficult to create a good quality of video without expensive equipments. Our system only needs one RGB camera to create a video and automatically overlays the effects of augmented reality (AR). In our system, the desktop server puts annotations on the objects and sends a video sequence to the clients. We used a laptop and smart phone as clients and they can only receive the data. Our system supports group communication across devices.

AR and virtual reality (VR) are often used for entertainment purpose. Sekai Camera (World Camera) [3] is a smart phone application which overlays meta-information (text, image, video) on the captured image. Camera pose estimation is required to track the target object. Especially for the interactive application, the algorithm has to be robust to viewpoint changes and occlusion in the case a user rotates and occludes the object by hands. In this paper, we propose vision-based 3D object tracking. Our experimental results show that our algorithm is robust to viewpoint changes and occlusion during interaction.

Laval Virtual VRIC'14, April 9-11, 2014, Laval, France.

This paper proposes a new system of sharing 3D object via networks with multiple clients using vision-based object tracking method. It helps users to communicate over 3D object and the augmented image with easy setups. Our method only uses one RGB camera for tracking and its algorithm is robust to viewpoint changes and occlusion during interaction by hands.

Section 2 refers to the related works of content sharing applications and tracking methods. Section 3 describes the framework of our system. Section 4 shows the experimental results and findings in our tracking algorithm.

2. RELATED WORKS

In this section, we refer to the related works of contents sharing applications and tracking methods.

There are many applications of VR contents sharing. HyperMirror [4] is a video sharing system to improve a communication environment. Both local and remote participants appear together on a shared video and they can feel as if they were in the same room. Watanabe et al. [5] proposed a learning support system with motion capture cameras and a RGB camera. The motivation of this system is as same as HyperMirror, but the size of the human rendered in the shared video is controlled by motion capture cameras. In Maimone et al.'s work [6], the clients can share the same view as the server. Their system tracks a gaze direction of a user for rendering the virtual world. Sheppard et al. [7] proposed a video sharing system, Virtual Room. They reconstructed human's motions by one grey camera and two RGB cameras and rendered them into a virtual room.

In those systems [4, 5, 6, 7], 3D position and pose of the target objects are needed for the interactive object sharing. In this paper, we propose to employ a vision-based object tracking method for avoiding the complex system configuration using multiple cameras with optical sensors [4, 6, 7], magnetic positioning sensors [5], etc.

For vision-based object tracking, SIFT [8] and SURF [9] are widely used as keypoint extractors and descriptors. They have scale, rotation and translation invariance, but they are weak at affine transformation. When these features are used for keypoint extraction, they cannot do correct matching because of it. Lepetit et al. [10] proposed a keypoint matching method using randomized trees learning. Their method is robust to viewpoint changes, but it costs time for learning and it is difficult to decide the threshold and parameters. Yoshida et al. [11] proposed a planar tracking method using view generative learning. Yoshida et al.'s method is also robust to viewpoint changes. Our proposed system uses Yoshida et al.'s method and extends to 3D object.

3. FRAMEWORK OF OUR APPLICATION

3.1 Network Structure

Fig.2 shows the network structure of our system. The user of the server has a camera and a target object. The server machine tracks the object and renders the view due to computational complexity. Client only receives the image data from the server. The server can connect to the multiple users across devices so that they can group up and communicate over the same content. In our system, we used a desktop computer (Windows7 64bit Intel Corei7-2600 12.0GB) as the server machine and a smart phone (Sumsung Galaxy S) and laptop (Windows8 64bit Intel Core i5-337U 8.0GB) as the clients.

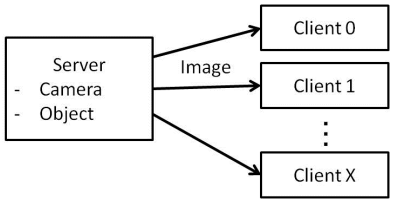


Fig.2 Network Structure

3.2 Tracking Algorithm

This section describes a vision-based tracking algorithm. We use the method of Yoshida et al. [11] and extend to 3D object. This method is robust to viewpoint changes and occlusion. The algorithm is shown in Fig.3. Learning phase is required in the offline phase to track and render the view in interactive time. This method is keypoint-based tracking and we use SIFT [8] for rotation and scale invariant feature.

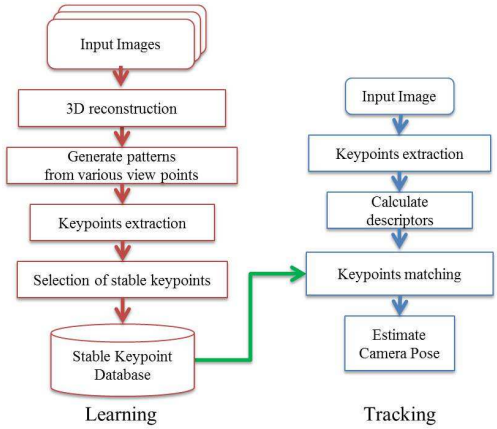


Fig. 3 Tracking Algorithm

3.2.1 Learning -- 3D Reconstruction

Our system needs a 3D model with texture. For this application, the models are reconstructed with Autodesk 123D Catch [12] because the reconstruction process is simple.

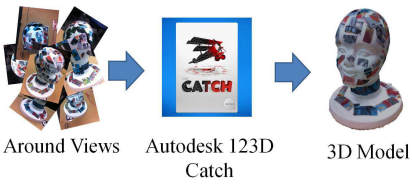


Fig.4 3D Reconstruction with Autodesk 123D Catch

3.2.2 Learning-- Generate Patterns from View Points

After reconstructing the model, the images from various viewpoints are generated virtually in OpenGL [13]. In this method, we assume that the lighting conditions are constant and the object is covered with Lambertian surface.

The camera pose is important for viewpoint settings. Since SIFT is scale invariant, there is no need to consider the distance from the object. Rotation matrix of the camera **R** is calculated with the equation (1). The parameter of ϕ represents the longitude, θ represents the latitude and ψ represents the camera spin. In this method, we use rotation invariant feature so we set the camera spin ψ as constant.

$$\mathbf{R} = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\phi & 0 & \sin\phi \\ 0 & 1 & 0 \\ -\sin\phi & 0 & \cos\phi \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{bmatrix} \quad (1)$$

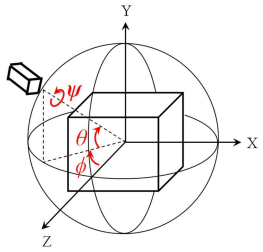


Fig. 5 Camera Pose and the Object

3.2.3 Learning -- Selection of Stable Keypoints

We extract the keypoints from the generated patterns by SIFT. Each keypoint p on the image is reprojected to p' in the 3D world coordinate by perspective matrix **P**. The equation is shown in equation (2).

$$p' \sim \boldsymbol{P}\, p \qquad (2)$$

Then we compare the Euclidean distance of the reprojected points from different views. If their Euclidean distance is close enough, these points are considered as the same point in 3D coordinate. The keypoints with high repeatability are called "stable keypoints" because they can be extracted from other viewpoint images. We sort the stable keypoints in order of repeatability and remain the top 1200. Each group of descriptors of the stable keypoint are clustered by k-means clustering (parameter: k) to represent the centroids.

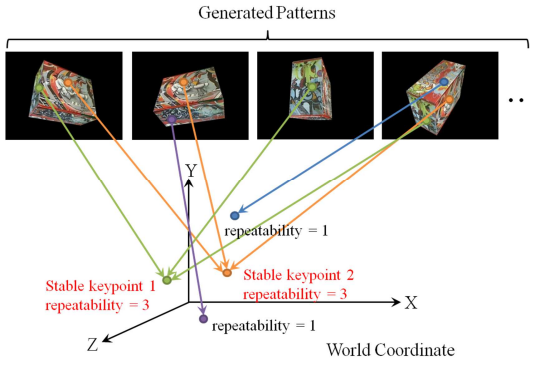


Fig. 6 Reproject Keypoints to 3D World Coordinate

4. EXPERIMENTS AND FINDINGS

This section shows the experimental results and findings of our tracking algorithm. We used the objects shown in Fig.7 for our experiments. The tracking results are shown in section 4.1. Section 4.2 discusses occlusion issue in terms of interaction. Section 4.4 discusses the limitation of our tracking algorithm.

4.1 Tracking Results

This section describes the tracking results with our algorithm. Fig. 8 shows our tracking results with **Head, Stadium** and **Cup Noodle**. We draw the boxes as sample annotations to demonstrate the accuracy of the camera pose estimation. These results show our method works and puts annotations on 3D objects with texture.

Matching results are shown in Fig.9 with **Box** and **Cylinder**. They show our algorithm is robust to viewpoint change in keypoint matching. We chose 151 images as sample data to calculate the corresponding matching rate with database. When parameter *k* in k-means clustering equals to five, the corresponding matching rate is 94%.

The comparison with randomized trees method [10] is shown in Fig. 10 and Fig.11 with **Head**. Our method performs better than the previous work in any angles.

4.2 Occlusion

This section discusses an occlusion issue in terms of interaction. When a user moves the object in front of the camera, the user's hands often occlude the target object.

Fig. 12 shows the tracking results during interaction. This user is trying to rotate or hold the object. Our system keeps tracking the object and putting annotations on the object with AR even if the user's hand hides the object.

Results are shown in Fig.8 with **Box** and **Cylinder**. These images show our method still has correspondent matching of the keypoints. This result explains that our method is working under partial occlusion.

4.3 Limitation of Our Method

This section describes the limitation of our method. We applied our system to an object with specular highlight. We chose a racing car in an outdoor environment for an example. The results of the tracking are shown in Fig. 11. In our method, the keypoints should be matched with more than three stable keypoints to estimate the camera pose. The success rate of this calculation was 49% out of 1496 frames in videos. Even if the pose was calculated, the estimation failed and the shape of the box collapsed as shown in Fig.11. It is because we assumed the object has Lambartian surface and the light condition is constant in one video sequence. However the car has specular highlight on its surface and the light condition changes in the outdoor environment. The area of the specular highlights is viewpoint dependent, so our method did not work with the car.

5. Conclusion

This paper proposes a new system of 3D Object sharing via networks with multiple clients. Our tracking algorithm is a vision-based 3D object tracking and robust to viewpoint changes and occlusion during interaction by hands. This system aims to support group communication over the same 3D object with AR annotations. Our future work is to track more than one object at the same time so that a user can talk over many objects with AR. In addition to it, we would like to improve our tracking algorithm to track a 3D object with specular highlight in an outdoor environment.

6. REFERENCES

[1] Youtube, http://www.youtube.com/

[2] Skype, http://www.skype.com/

[3] Sekai Camera, http://sekaicamera.com/

[4] O. Morikawa and T. Maesako. ", HyperMirror : toward pleasant-to-use video mediated communication system", In Proc. ACM conference on Computer Supported Cooperative Work, pp.149 - 158, ACM 1998.

[5] K. Watanabe and M.Yasumura ", Visual Haptics : generating haptic sensation using only visual cues", International Conference on Advances in Computer Entertainment Technology, pp.405-405, ACE 2008.

[6] A. Maimone and Henry Fuchs ", A first look at a telepresence system with room-sized real-time 3D capture and life-sized tracked display wall", In Proc. ACM conference on Computer Supported Cooperative Work, 2011.

[7] R.M.Sheppard, M.Kamali, R.Rivas, M.Tamai, Z.Yang, W.Wu and K.Nahrstedt ", Advancing interactive collaborative Mediums through Tele-immersive dance (TED): A symbiotic creativity and design environment for art and computer science", ACM MM 2008.

[8] D.G. Lowe ", Distinctive image features from scale invariant keypoints", International Journal of Computer Vision, 60, pp.91 -110. 2004.

[9] H. Bay, T.Tuytelaars and L.V.Gool ", SURF: Speed up robust features", European Conference on Computer Vision, pp.404-417, 2006.

[10] V. Lepetit and P.Fua ", Keypoint recognition using randomized trees", IEEE transactions on Pattern Analysis and Machine Intelligence, 28(9) pp.1465-1479, 2006.

[11] T. Yoshida, H.Saito, M.Shimizu, T.Taguchi “, Stable keypoint recognition using viewpoint generative learning", Proceedings of the international computer vision theory and applications, vol.2, pp.310 -315, Feb. 2013.

[12] 123D Catch, http://www.123dapp.com/catch

[13] OpenGL, http://www.opengl.org/

Acknowledgements

This work was partially supported by MEXT/JSPS Grant-in-Aid for Scientific Research(S) 24220004.

(a) Head

(b) Stadium

(c) Cup Noodle

(d) Box

(e) Cylinder

Fig.7 Target Objects

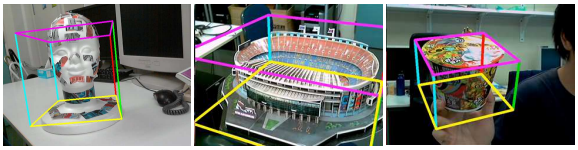
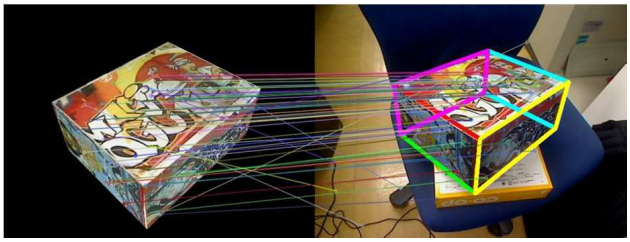
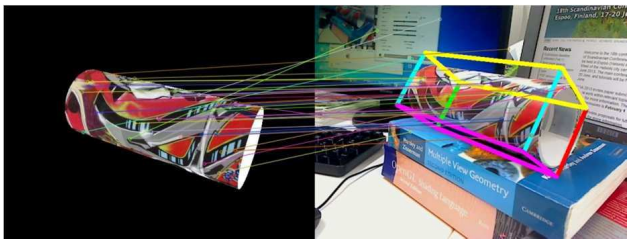


Fig.8 3D Object Tracking

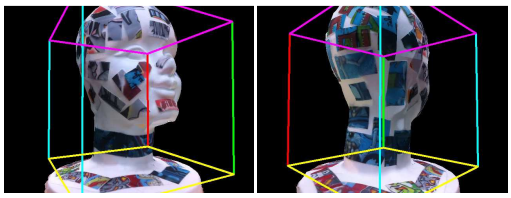


(a) Box

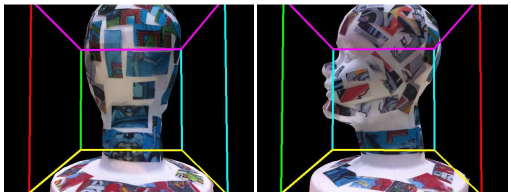


(b) Cylinder

Fig.9 Keypoint Matching



(a) 30 degrees **(b) 135 degrees**



(c) 180 degrees **(d) 270degrees**

Fig.10 Our Proposed Method

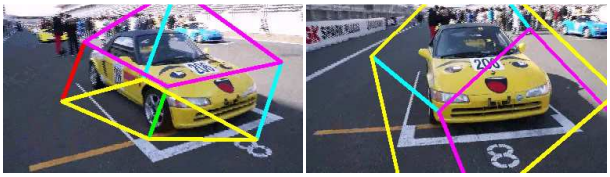
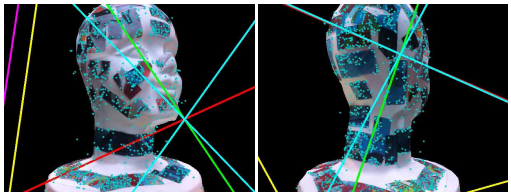
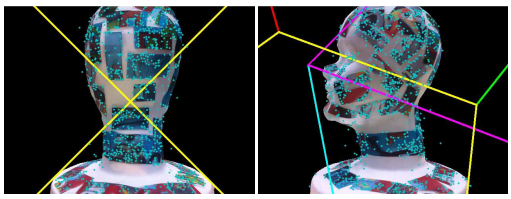


Fig.14 Tracking of the Car on Racing Circuit



(a) 30 degrees **(b) 135 degrees**



(c) 180 degrees **(d) 270 degrees**

Fig.11 Randomized Trees Method [10]

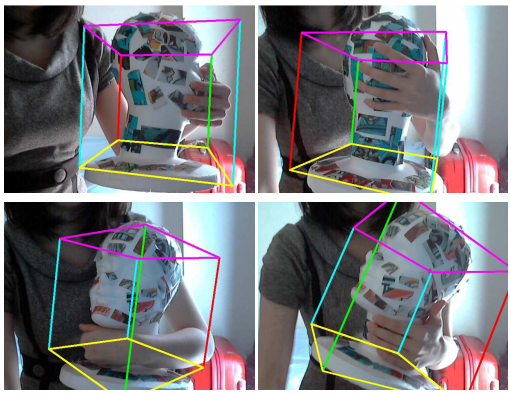
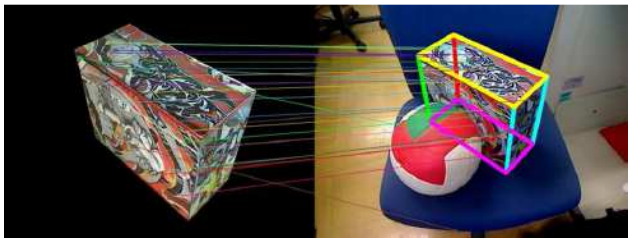
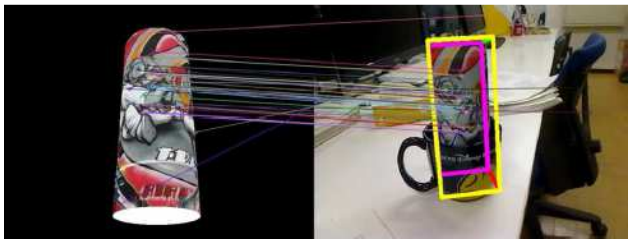


Fig.12 Occlusion during Interaction



(a) Box



(b) Cylinder

Fig.13 Matching Result in Occlusion