KICS The Korean Institute of Communications and Information Sciences

Available online at www.sciencedirect.com

ScienceDirect

ICT Express 7 (2021) 71–75

www.elsevier.com/locate/icte

3D mesh transformation preprocessing system in the real space for augmented reality services

Young-Suk Yoon[a], Sangwon Hwang[b], Dogyoon Lee[b], Sangyoun Lee[b], Jae-Won Suh[c], Sung-Uk Jung[a,*]

[a] *Content Research Division, Electronics and Telecommunications Research Institute, Daejeon, Republic of Korea*
[b] *Department of Electrical and Electronic Engineering, Yonsei University, Seoul, Republic of Korea*
[c] *College of Electrical and Computer Engineering, Chungbuk National University, Chungbuk, Republic of Korea*

Received 30 November 2020; received in revised form 27 January 2021; accepted 3 February 2021
Available online 12 February 2021

Abstract

We propose a preprocessing system that transforms the real world into 3D mesh virtual world to provide augmented reality (AR) services. The proposed system uses a monocular RGB camera to obtain sequential images to generate real space. First, we create 3D real-world space composed of point clouds using sequential color images. And then, we segment the objects in each color image and assign an identification number for each object. Also, we execute '3D point labeling', which assigns identification numbers from 2D segmented object to 3D point cloud through a simple projection manner with several parameters. This process could collect 3D point cloud with same the label by as an object. Finally, only the 3D point clouds that have the same identification number by as an object are used to generate 3D mesh. This paper confirmed that 3D mesh could be created using the only general monocular camera without the help of special 3D sensors.

Keywords: 3D mesh; 3D point cloud; 3D point labeling; AR service

1. Introduction

As developed wireless communication with 5G technology, application services expand into extensive capacity contents related to 3D space. Several market forecasting agencies argue that 'Augmented Reality (AR)' is one of the future growth engine technology that will lead the next five years. Additionally, recently AR contents in a real-world environment have been evolving to allow users to participate directly and interact among users, virtual objects, and real objects. To utilizing these contents, sophisticated modeling of real space should be satisfied that virtual objects are express as realistically to users could feel a sense of real immersion.

For providing natural AR services, Apple and Google provide functions that allow users to add depth information in ARKit [1] and ARCore [2], respectively. ARKit of Apple incorporation can provide pixel-level depth information from LiDAR (Light Detection and Ranging) scanner, which is mounted on newly released products in 2020. Fig. 1 shows an example of ARKit that recognizes real space with LiDAR sensor. Also, ARCore of Google incorporation focuses on estimating depth images based on deep neural network using artificial intelligence technology to connect and augment the surrounding environment with virtual objects.

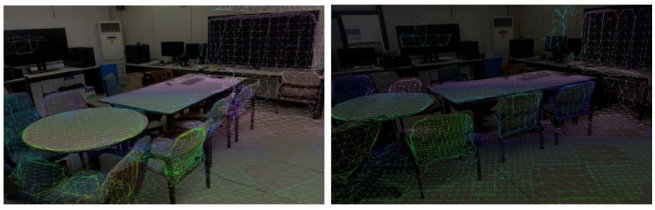


Fig. 1. Example of ARKit [1].

Our proposed methods of 3D mesh transformation preprocessing system are as follows. First, we introduce our processes in order of system flow. Specifically, '3D point labeling' process describes how we utilize 2D segmented images to

* Corresponding author.
E-mail addresses: ys.yoon@etri.re.kr (Y.-S. Yoon), sangwon1042@yonsei.ac.kr (S. Hwang), nemotio@yonsei.ac.kr (D. Lee), syleee@yonsei.ac.kr (S. Lee), sjwon@cbnu.ac.kr (J.-W. Suh), brcastle@etri.re.kr (S.-U. Jung).

Peer review under responsibility of The Korean Institute of Communications and Information Sciences (KICS).

https://doi.org/10.1016/j.icte.2021.02.001

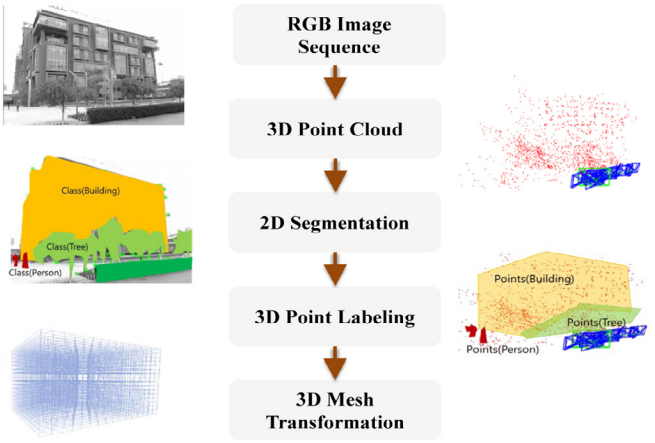


Fig. 2. Block diagram of proposed system.

assign labeling information on a 3D point cloud. As providing our results of transformed 3D mesh using the sequential RGB images, we show how effectively our system is applied to AR contents. Finally, we summarize the conclusions.

2. Proposed system

Fig. 2 shows the overall block diagram of the proposed system in this paper. As shown in Fig. 2, the middle block explains the process procedure for the proposed paper. Also, the left and right in Fig. 2 stand for the examples of the application of each block.

The first suggested method is to prepare an image sequence using an RGB camera and use it as an input to our proposed system. The upper left-hand figure in Fig. 2 illustrates an example of an RGB image sequence acquired with a monocular camera. Multiple cameras may be used to obtain RGB image sequences of various channels for accurate spatial recognition. Moreover, a variety of additional sensors that can provide depth and location information may be available. However, by only using single-sided lenses of personal mobile devices capable of acquiring general RGB images, we obtain real-world information to provide augmented reality services.

Secondly, the process of creating a 3D point cloud by utilizing RGB images acquired earlier. Fig. 2 shows that the images (upper left-hand in Fig. 2) are used to generate 3D point cloud (upper right-hand in Fig. 2). Our system has expanded based of ORB-SLAM algorithm [3] among various methods to generate 3D space. The extended method stores and reads 3D spatial information so that it can be reused, and has been optimized and lightened to work on mobile devices as well as PCs. This method can simultaneously generate 3D space and localize the current camera pose in the generated map with relatively light ORB feature descriptors [4].

Thirdly, we do not segment 3D point cloud directly but 2D RGB images due to the sparsity of 3D point cloud. In the proposed system, the YOLACT algorithm [5] is used to segmented objects at class level (Instance Segmentation). The YOLACT method creates a set of masks for the prototype and performs two-stage simultaneously, which calculates mask coefficients per instance. Thus, it is possible to deduce fast

Fig. 3. An input image & instance segmented results by YOLACT.

and relatively accurate object instance segmentation results. Fig. 3 shows the laboratory images used in our experiment and the results of segmented objects by instance in different colors with YOLACT algorithm, respectively. The 2D image-based instance object segmentation uses RGB images from the real world independently. Therefore, the class or identification number of object information can be classified differently even it is the same object in the real space.

3. 3D point labeling

The next fourth involves the process of '3D point labeling', which assigns of segmented objects in a 2D RGB image to 3D point cloud in 3D space. In the global 3D real space, each object can be assigned one class and one identification number. However, in a 2D image in which 3D points are projected, it is difficult to simultaneously recognize the class and identification number of objects across the real world because they have only local partial information. To solve this problem, objects appearing in RGB images that were used to model real space must be identified as the same object, which means that each class and identification number should be connected. Therefore, we connect the segmented information of each object in the proposed system like the Re-Identification (Re-ID) problem.

First, we compute to estimate homography matrix H. Fig. 4 shows the 0th (a) and 10th (c) images of RGB sequences used in our experiment, respectively. Fig. 5 also expresses the process of estimating H with (a) and (c) of Fig. 4. We obtain N matching pairs $P_1 = \left\{[x_1^n, y_1^n, 1]^T | n = 1, 2, \ldots, N\right\}$ and $P_2 = \left\{[x_2^n, y_2^n, 1]^T | n = 1, 2, \ldots, N\right\}$ from the two RGB images using the SURF descriptor [6]. And then, homography matrix H is estimated using Random Sample Consensus (RANSAC) and Least Square [7] approaches. We can obtain warped image $I_{1\to 2}$ as shown in below Eq. (1) by converting the pixel coordinates in homogeneous method, which is appending 1 to the last dimension ($I = [u, v, 1]^T$).

(a) 0^{th} image (b) 0^{th} image with image warping (c) 10^{th} image

Fig. 4. 0th image, warped 0th image, and 10th image.

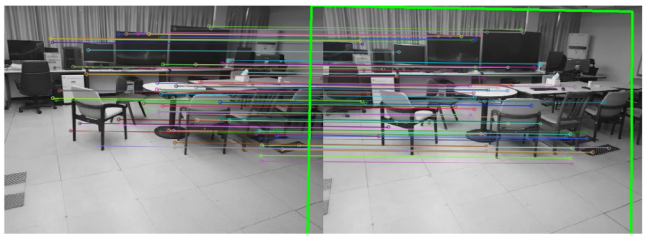


Fig. 5. Estimating homography matrix.

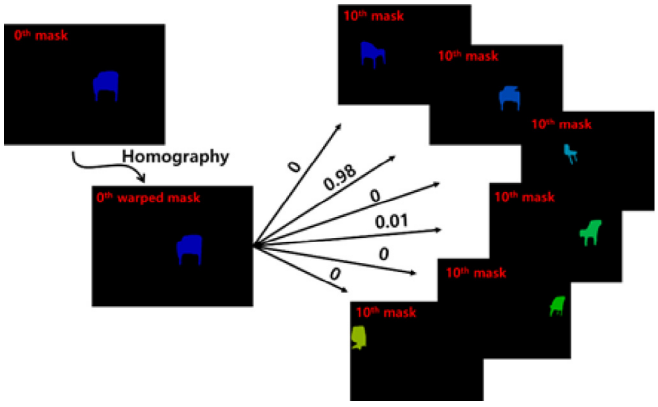


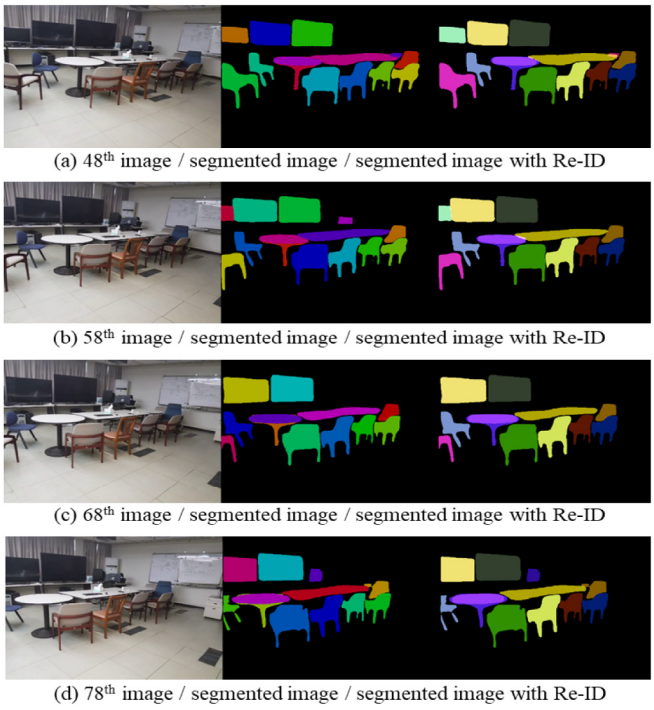
Fig. 6. Connecting and transferring process of object information.

$$I_{1\to 2} = H I_1 \tag{1}$$

Fig. 4 (b) stands not only Eq. (1) but also the warped image with homography matrix H applied to Fig. 4(a). In fact, estimating homography assumes that the space is the plane space, theoretically, which may not perfectly apply to AR services that target to interact 3D real space. However, when the depth values of all points are accurately unknown, and the difference in the temporal and spatial difference of each RGB image is small, 3D space could be approximate 2D planar space, which our experiment proves qualitatively.

Then, using the estimated homography matrix between images, masks with object segmented information by YOLACT [5] are also warped in the same way. Comparing the Intersection over Union (IoU) between the segmented object masks, we pass on the class and identification number of the allocated object to the following image. Our proposed system actively utilizes key frames of the ORB-SLAM, which is used to generate 3D point cloud, to reduce the burden on our system. In addition, when the frame distance between key frames is quite far, our system was designed to adding an intermediate pseudo key frame with a fixed distance between frames to transfer labels efficiently.

Fig. 6 illustrates the connecting and transferring process of object information. The left images of Fig. 6 show the segmented chair (0th mask) and the warped segmented chair (warped 0th mask by H). Firstly, we set all identification numbers to '−1' and 0th of chair mask to '0'. 0th chair mask is warped using the homography matrix H, which mention in Eq. (1). And then, we compute IoU between the warped 0th mask and segmented masks for each chair in 10th image.



(a) 48th image / segmented image / segmented image with Re-ID

(b) 58th image / segmented image / segmented image with Re-ID

(c) 68th image / segmented image / segmented image with Re-ID

(d) 78th image / segmented image / segmented image with Re-ID

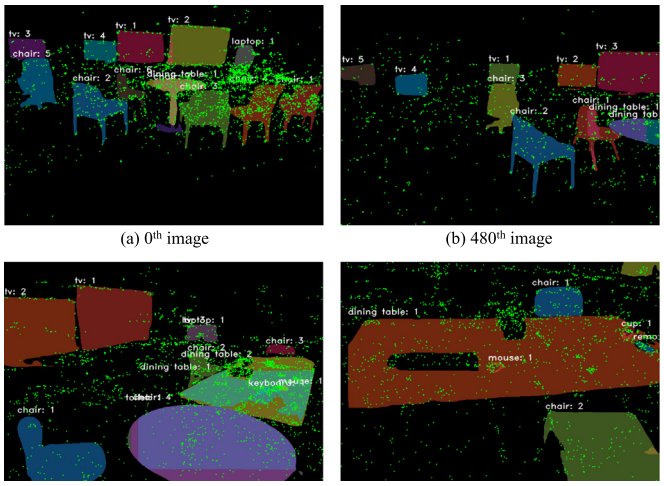
Fig. 7. RGB Image, Instance Segmentation, Re-ID.

Our system determines that the mask with the largest value of threshold (0.6) or higher is the same object and passes on the identification number of that object to it. The second chair mask of 10th image has the largest IoU value (0.98) on the right images of Fig. 6, indicating that the same chair to 0th chair mask. When the IoU is less than threshold, the proposed system determine that it is a new object and assigns a new identification number. The class of objects is already known, comparing only the same class reduce the computational cost. This process allows the proposed system to associate a sequence of 2D images with an identification number to each object as if Re-ID was applied. Fig. 7 shows the results that when there is a connection between segmented objects or not.

3D point cloud is projected into 2D segmented images with Re-ID applied to assign classes and identification numbers to 3D points for each object from them. Fig. 8 depicts a visualized image by projecting all 3D points after the Re-ID connecting segmented object information. Through the proposed '3D point labeling', we can finally label and transfer both the class and identification number of each object on its 3D points.

4. 3D mesh transformation

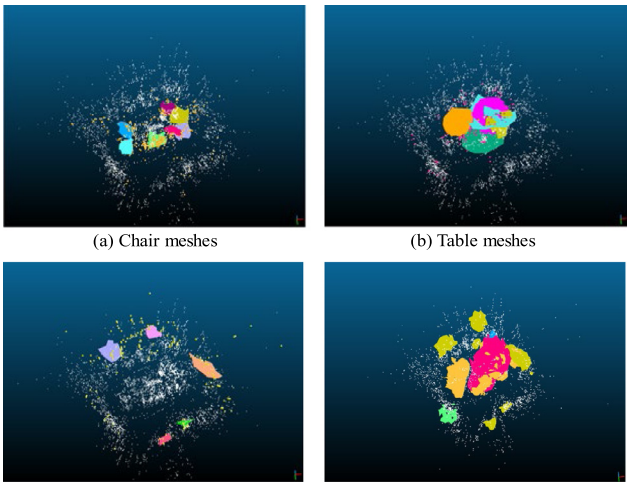
Finally, as a result of the 3D point labeling process, points are gathered for each object and converted into a 3D mesh for each set of points for each object. Not only geometric information of each point but also normal vector information for each triangular mesh formed by the points is required in order to represent a 3D mesh with points. It is difficult for 3D points without a normal vector information to accurately



(a) 0^{th} image (b) 480^{th} image

(c) 2106^{th} image (d) 2326^{th} image

Fig. 8. Projected 3D points on segmented images.



(a) Chair meshes (b) Table meshes

(c) TV meshes (d) All meshes

Fig. 9. Screenshot of each class meshes.

visualize because the inside and outside of the 3D mesh cannot be distinguished.

The 3D point cloud generated from the 2D RGB image sequence used in the experiment has a low density and does not have a normal vector. Therefore, the normal vector of the plane including 3D points is approximated by using the geometric coordinates of the neighboring points of the point to which the normal vector is to be predicted. We use not only the predicted normal vector values but also geometry information (x,y,z) together. And then, we apply the Poisson reconstruction [8] algorithm to the 3D mesh of the object's surface. Finally, our proposed system places the 3D meshes with reconstructed surfaces for each object on a 3D point cloud.

5. Experimental results

The information of both a sequence of 2D RGB images and a 3D point cloud generated from it used for the experiment in this paper are as follows. The sequence of RGB images consists of a total of 2702 images, and the resolution of the image is 640 × 480 of Video Graphics Array (VGA) scale. The experimental environment of the acquired real space is indoor. There are tables, chairs, TVs, keyboards, laptops, and other objects. The 3D point cloud obtained by applying the revised ORB-SLAM [5] algorithm to the experiment sequence has 5870 3D points. The geometric information of each point has (x,y,z) positional values in the world coordinate system.

Fig. 9 shows (a) chairs, (b) tables, and (c) TVs, which are represented by 3D meshes created as individual objects with different colors through the MeshLab [9]. Moreover, Fig. 9(d) represents the five main objects, such as chairs, tables, TVs, keyboards, and laptops, in different colors for each class by using MeshLab. As can be seen from the experimental sequence of RGB images, the real space in which chairs are located around the table can be seen well in Fig. 9(d). ARKit using the lidar sensor shown in Fig. 1 generates roughly one 3D mesh, but the proposed method has the advantage of generating a mesh for each individual. Moreover, although the proposed method has a slight delay, it can process an average speed of 8.06fps per input image in one Geforce GTX 1080Ti GPU and i7-6700 CPU environment. If the implementing technique of the proposed method is optimized, there is room for speed improvement.

6. Conclusion

In this paper, we proposed a method of preprocessing and generating a real space with 3D meshes on a 3D point cloud in order to provide an interactive AR service by processing a sequence of 2D RGB images that can be obtained with not any special sensor but a general monocular RGB camera. The proposed system recognized the information of an instance-level object as '3D point labeling' that applies image segmentation algorithms to 2D images, estimates homography matrices, applies image warping, and projects 3D points to the segmented images. Then, our system gathered 3D points for each object and transformed them into 3D meshes on 3D point cloud. The final experimental results confirmed that 3D meshes could be produced in the form of each object in the real space.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research was supported by Ministry of Culture, Sports and Tourism (MCST), South Korea and Korea Creative Content Agency (KOCCA) in the Culture Technology (CT) Research & Development Program. [R2018030392, Development of participational AR platform for large-scale cultural space].

References

[1] https://developer.apple.com/augmented-reality/arkit/.

[2] https://developers.google.com/ar.

[3] Raul Mur-Artal, Jose Maria Montiel, Juan D. Tardos, ORB-SLAM: a versatile and accurate monocular SLAM system, IEEE transactions on robotics 31.5 (2015) 1147–1163.

[4] Ethan Rublee, et al., ORB: An efficient alternative to SIFT or SURF, in: ICCV, 2011.

[5] Daniel Bolya, Chong Zhou, Fanyi Xiao, Yong Jae Lee, YOLACT: Real-time instance segmentation, in: ICCV, 2019.

[6] Herbert Bay, Tinne Tuytelaars, Luc Van Gool, Surf: Speeded up robust features, in: ECCV, 2006.

[7] Ruwen Schnabel, Roland Wahl, Reinhard Klein, Efficient RANSAC for point-cloud shape detection, in: Computer Graphics Forum. Vol. 26, (2) Blackwell Publishing Ltd, Oxford, UK, 2007.

[8] Michael Kazhdan, Matthew Bolitho, Hugues Hoppe, Poisson surface reconstruction, in: Proceedings of the Fourth Eurographics Symposium on Geometry Processing, Vol. 7, 2006.

[9] https://github.com/cnr-isti-vclab/meshlab.