PlanAR: Accurate and Stable 3D Positioning System via Interactive Plane Reconstruction for Handheld Augmented Reality

Ami Miyake, Hideaki Uchiyama[a], Atsushi Shimada and Rin-ichiro Taniguchi[b]
Kyushu University, Fukuoka, Japan

Keywords: 3D Positioning, vSLAM, Handheld Augmented Reality, Interactive 3D Reconstruction.

Abstract: This paper presents a ray-casting-based three-dimensional (3D) positioning system that interactively reconstructs scene structures for handheld augmented reality. The proposed system employs visual simultaneous localization and mapping (vSLAM) technology to acquire camera poses of a smartphone and sparse 3D feature points in an unknown scene. First, users specify a geometric shape region, such as a plane, in captured images while capturing a scene. This is performed by manually selecting some of the feature points generated by vSLAM in the region. Next, the system computes the shape parameter with the selected feature points so that the scene structure is reconstructed densely. Subsequently, users select the pixel of a target point in the scene at one camera view for 3D positioning. Finally, the system computes the intersection between the 3D ray computed with the selected pixel and the reconstructed scene structure to determine the 3D coordinates of the target point. Owing to the proposed interactive reconstruction, the scene structure can be estimated accurately and stably; therefore, 3D positioning will be accurate. Because the geometric shape used for the scene structure is a plane in this study, our system is referred to as PlanAR. In the evaluation, the performance of our system is compared statistically with an existing 3D positioning system to demonstrate the accuracy and stability of our system.

1 INTRODUCTION

Three-dimensional (3D) positioning refers to the computation of the 3D coordinates of a target point in a scene (Polvi et al., 2016). The applications based on 3D positioning includes, but not limited to, placing 3D annotations in augmented reality (AR) (Wither et al., 2009; Marchand et al., 2016), manipulating 3D objects in virtual reality (VR) (Baillot et al., 2001; Jung et al., 2002; Henrysson et al., 2005), and distance measurement for surveying systems[1]. Compared with measuring-tape- or laser-range-finder-based approaches (Amann et al., 2001), visual–inertial methods are easy to use, reasonably accurate, and practical in various situations because they can be implemented on mobile devices (Polvi et al., 2016; Tashiro et al., 2019). Furthermore, they are used in a noncontact manner when keeping users away from a target point.

As implemented in ARKit by Apple or ARCore by Google for placing AR annotations via 3D positioning, one approach is to use ray casting (Hinckley et al., 1994) derived from object manipulation in VR (Baillot et al., 2001). To determine the 3D coordinate of a target point in the scene, selecting its pixel at one camera view is sufficient if the scene structure is completely known such as VR systems. For AR systems, the procedure is composed of three steps, as follows: With visual simultaneous localization and mapping (vSLAM) (Taketomi et al., 2017), sparse sets of 3D feature points are reconstructed in a scene while estimating camera poses. Next, the feature points are used to estimate geometric shapes such as planes and spheres. This process is equivalent to generating a complete scene structure. Subsequently, 3D positioning can be performed. When users select the pixel of a target point at one camera view, the intersection between the 3D ray of the selected pixel and the scene structure is computed to determine the target 3D coordinate. The final process is known as ray casting.

The drawback of the existing ray-casting-based methods is that the results of 3D positioning is neither accurate nor stable owing to the inaccuracy of the reconstructed scene structure. In other words, both the accurate scene structure and pixel selection of a

[a] https://orcid.org/0000-0002-6119-1184
[b] https://orcid.org/0000-0002-2588-6894
[1] http://armeasure.com/

Miyake, A., Uchiyama, H., Shimada, A. and Taniguchi, R.
PlanAR: Accurate and Stable 3D Positioning System via Interactive Plane Reconstruction for Handheld Augmented Reality.
DOI: 10.5220/0008942307830791
In *Proceedings of the 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2020) - Volume 5: VISAPP*, pages 783-791
ISBN: 978-989-758-402-2; ISSN: 2184-4321

target point are necessary for accurate 3D positioning. When using automatic reconstruction methods, several heuristic parameters should be set internally to determine the structure size, neighborness of 3D points, and tolerance to noise (Schnabel et al., 2007). However, optimizing the aforementioned parameters for various scene structures is difficult in practice. Since these parameters need to be determined in advance, the estimated scene structure is not always accurate; i.e., one planar surface can be divided into two planes, or a stepped surface can be detected as one plane. This inaccuracy largely degrades the usability of the existing systems.

We herein propose a simple but effective interactive reconstruction method for achieving accurate and stable ray-casting-based 3D positioning systems. The proposed system employs a vSLAM technology to acquire camera poses of a smartphone and sparse 3D feature points in an unknown scene. First, users specify a geometric shape region, such as a plane, in a scene by selecting the pixels of the feature points generated by vSLAM in the region. This can be performed easily by running a fingertip along the region on the smartphone screen. Next, the system computes the shape parameter with the selected points such that the scene structure is reconstructed densely. Subsequently, users select the pixel of a target point at one camera view for 3D positioning. Finally, the system computes the intersection between the 3D ray computed with the selected pixel and the scene structure to determine the coordinate of the target point. Owing to our user interaction-based reconstruction, the scene structure can be estimated accurately and stably with little user effort. This process is invaluable for achieving accurate and stable 3D positioning even though it is simple and straightforward. As explained in Section 3, our system is referred to as PlanAR because we focus on generating planes as geometric shapes. Nonetheless, we can easily incorporate other shapes, such as spheres and cylinders, into our system. In the evaluation, the performance of our system is compared statistically with an existing system to demonstrate the effectiveness of our system. It is noteworthy that we assumed that the scene contained some textures to produce feature points for vSLAM.

2 RELATED WORK

Camera-based approaches for 3D positioning can be divided into two categories. The first approach is to use ray casting against a densely reconstructed scene structure, as explained in Section 1. By using dense point-cloud reconstruction (Whelan et al., 2016; Fuhrmann et al., 2014) or geometric shape reconstruction from sparse feature points (Roberto et al., 2017), ray-casting-based 3D positioning at one camera view can be performed (Nuernberger et al., 2016; Lien et al., 2016). The drawback of this approach is that the inaccuracy of automatic scene reconstruction degrades the accuracy of 3D positioning. This is generally caused by certain heuristic parameters used in the plane reconstruction method (Schnabel et al., 2007). Owing to inaccurate scene structures, 3D positioning becomes inaccurate consequently.

The second approach is to use triangulation with multiple camera views (Polvi et al., 2016; Tashiro et al., 2019). This approach is also based on a vSLAM technology. To compute the 3D coordinate of a target point in a scene, the pixel selection of a target point is required from at least two views such that the triangulation technique can be applied (Hartley and Sturm, 1997). To allow users to easily select pixels at two camera views, Polvi *et al.* proposed the use of an epipolar geometry (Zhang, 1998) to provide auxiliary visualization (Polvi et al., 2016). In this system, users first select the pixel of a target point at one camera view and repeats it on the visualized epipolar line at the other view. Tashiro *et al.* proposed an alternative system such that the users can select pixels at any number of camera views to accommodate noisy user inputs (Tashiro et al., 2019). In this system, techniques on multiple view geometry are incorporated to improve the accuracy and stability of 3D positioning (Rumpler et al., 2011; Hess-Flores et al., 2014; Yu and Gallup, 2014). This approach is generally more accurate than ray-casting-based approaches because it does not depend on automatic structure reconstruction. However, it requires more operation time because selecting pixels at many camera views becomes more stressful when the number of target points to be measured is increased.

The comparison of the first, second, and our proposed approaches for 3D positioning is presented in Table 1. To compute the 3D coordinate of one target point in a scene, the number of required operations varies according to the approaches. Our system is the extension of the first approach. Particularly, we focus on achieving an accuracy similar to that of the second approach while maintaining the operation time of the first approach. Because our proposed interactive reconstruction is simple, it can be performed quickly. Therefore, the operation time does not increase significantly while accuracy is maintained, as discussed in Section 4.

Table 1: Comparison of approaches for 3D positioning.

Approach	Accuracy	Num. of operations
Ray casting with automatic reconstruction (first approach)	Not stable	1
Interactive triangulation with multiple views (second approach)	Stable	more than 2
Ray casting with interactive reconstruction (IR) (Proposed)	Stable	1 after IR

3 PROPOSED SYSTEM

3.1 Overview

Figure 1 illustrates the flow of our proposed system. The processes are divided into user and system sides.

After starting the system, vSLAM is performed consistently as a background process such that camera poses and sparse 3D map points can be acquired continuously. We used ARCore as the vSLAM library. First, map points generated by vSLAM are visualized on the camera images. Next, users select a geometric shape region in the images to densely reconstruct the scene structure. This is performed by selecting the pixels of the map points detected in the region. In this study, we focus on using planes. After pixel selection, the plane parameter is estimated from the selected map points. Users can reconstruct any number of small or large planes in the scene, if necessary. Finally, users can select the pixels of target points on the reconstructed planes for 3D positioning.

It is noteworthy that the 3D coordinates are estimated in the vSLAM coordinate system, which is typically determined by vSLAM. This coordinate system is sufficient for AR annotation placement. However, the 3D coordinates are not comprehensive for users because the vSLAM coordinate system is generally unknown. Therefore, our prototype system is designed to measure the distance between two points, as described in Section 4.

3.2 Visualization of Map Points

Recent visual-inertial SLAM can generate sparse 3D map points for an accurate camera pose estimation (Delmerico and Scaramuzza, 2018), as implemented in ARCore. Since ARCore provides some map points in the scene, we extract them from ARCore at every frame, and visualize them on the images, as illustrated in Figure 2(a). To visualize them, a fixed size of a sphere, such as a radius of 1 cm, is placed at each map point.

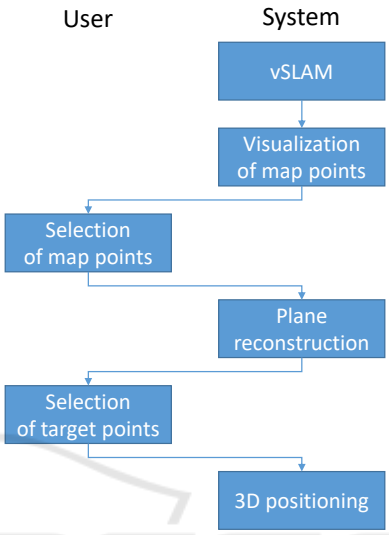
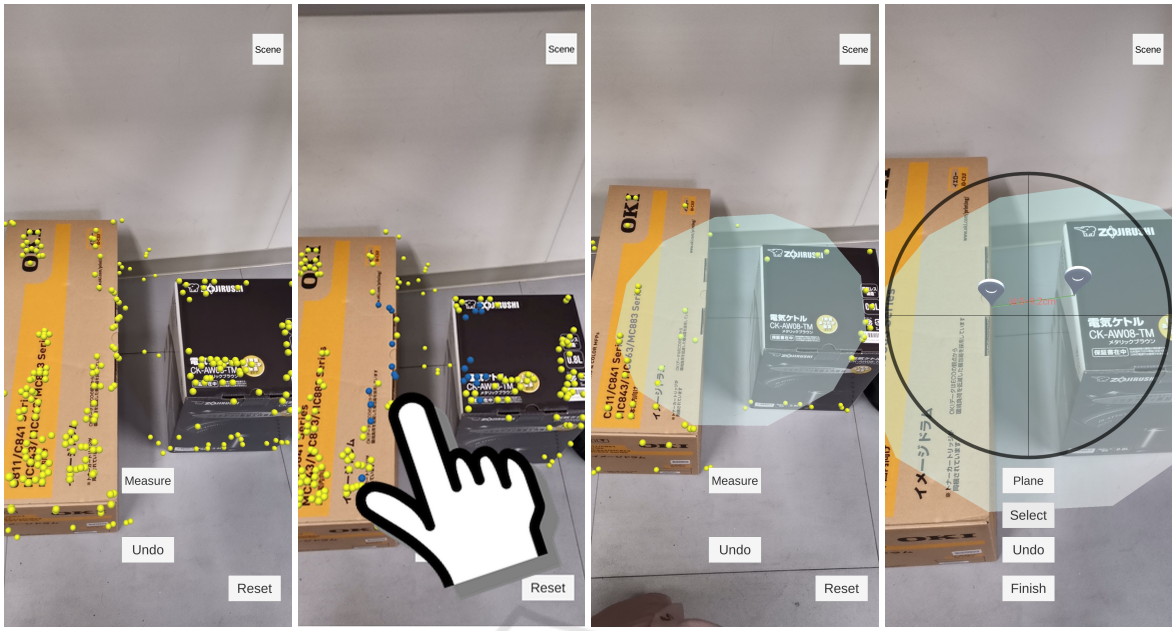


Figure 1: Flow of our proposed system. The processes are divided into user and system sides. Users select pixels for the interactive reconstruction and 3D positioning of target points. While executing vSLAM, the system reconstructs the scene structure from the map points selected by users and computes the 3D coordinates of the target points.

3.3 Selection of Map Points

To reconstruct the scene structure with a plane, users are required to select the pixels of map points detected on a plane. If users are required to precisely select the point individually, this process would be laborious. However, a few points are sufficient to compute the plane parameter; at least three points are necessary. Therefore, this process can be performed easily if map points are detected on plane regions.

In our system, users are asked to run a fingertip along the plane region, as illustrated in Figure 2(b). In this figure, the user attempts to create a plane on the top of two boxes, which height is same. The color of the selected map points is blue, whereas that of unselected ones is yellow. To detect whether a map point is selected, ray casting from a selected pixel to unselected map points is performed. Even though users may select the points that do not belong to a plane, such points can be removed as outliers in the process of robust estimator-based plane reconstruction, as explained in Section 3.4. Therefore, users can perform



(a) Visualization of map points (b) Selection of map points (c) Plane reconstruction (d) Selection of target points

Figure 2: Each process in our proposed system. Map points from vSLAM are first visualized and then selected by users to reconstruct a plane. Subsequently, users select the pixels of the target points on the reconstructed plane to measure the distance between two points.

this process in a friendly manner.

3.4 Plane Reconstruction

After users have selected the map points on a plane, the plane parameter is computed using the points. This process is based on principal component analysis (PCA). PCA can compute two axes on a plane and one axis perpendicular to the plane using 3D map points. To remove outliers, RANSAC is applied as a robust estimator (Fischler and Bolles, 1981).

When using PCA, the direction of two axes on a plane is determined according to the arrangement of the map points. This means that it is difficult to consistently estimate the direction such that one axis always faces the same direction. Therefore, a plane is visualized using a polygon or a circle, as illustrated in Figure 2(c). The polygon center is located at the center of the selected map points, and its radius is determined from the center to the farthest point.

3.5 Selection of Target Points

For 3D positioning, users select the pixel of any target point on the plane. The target point in this process is not equivalent to a map point generated by vSLAM.

The process of computing the 3D coordinates of a target point is as follows. From vSLAM, camera poses are acquired when tracking is successful. When users select the pixel of a target point in the image, the 3D ray of the pixel is computed using a camera pose and pixel location in the image, as well as VR systems (Baillot et al., 2001). To support an accurate pixel selection, we used a reticle marker, as implemented in SnipAR (Tashiro et al., 2019)(see Figure 2(d)). Subsequently, ray casting is performed from the pixel to the plane in the scene. Finally, the intersection between the plane and 3D ray is computed as the 3D coordinates of the target point.

4 EVALUATION

4.1 Evaluation Design

To evaluate the performance of our proposed system, we developed an application to measure the distance between two points in a scene, such as a commercial AR ruler application. When using the application, users are required to perform 3D positioning twice to measure the distance. In this evaluation, the participants were asked to measure the size of some objects.

As illustrated in Figure 3, the target scene for the evaluation comprised several objects that varied by size and shape and placed on a mesh-type metal rack board. Compared with simple environments used in

Figure 3: Evaluation scene in a storeroom. The task was to measure the width of objects stored on a rack board.

other evaluations (Polvi et al., 2016; Tashiro et al., 2019), we selected an actual environment in a storeroom because we focused on investigating the performance in practical use.

In our preliminary experiments, we discovered that ARcore could not reconstruct a plane on a rack board even though it generated some map points. In practical situations, automatic plane detection did not work or failed. Our interactive plane reconstruction is highly useful in such situations because the plane can be stably reconstructed with map points selected by users, as discussed in Section 5. As a benchmarking method, we used SnipAR because its accuracy in 3D positioning is better and it is also less time-intensive than other methods (Tashiro et al., 2019).

4.2 Task

The task was to naively measure the distance between two points using two systems: our proposed PlanAR and SnipAR. Particularly, the participants were asked to measure the width of five objects on a rack board. In practice, this process is equivalent to quantitatively measuring the occupancy on each rack board at a distance in a non-contact manner.

Before the evaluation, we provided detailed instructions of how each system is to be used and provided sufficient time for trial use. The following instructions were provided.

1. Participants must measure the distances as accurate as possible.
2. Participants assess whether to re-perform 3D positioning, at their own will.
3. Participants start the task 1 meter away from the rack.
4. After the participants have started the task, they can move freely.
5. Participants push the finish button when they are satisfied with the measurement results.
6. For PlanAR, participants first create a plane on a rack board and then perform 3D positioning.

To clarify the order of the distance measurement for the participants, the order was visualized with small-numbered tags in the scene.

The evaluation procedure is as follows:

1. pre-questionnaires regarding participant background,
2. instructions of how to use each system with an instruction movie and trial uses,
3. task of measuring the distances between five sets of two points using each system,
4. and post-questionnaires for feedback regarding the usability of each system.

For each participant, the order of the task for the two systems was randomized to be counterbalanced. Each participant required approximately 30 min to complete all procedures.

4.3 Evaluation Criteria

The first criterion was the operation time required for the distance measurement. The time measurement started when the participants pushed the start button. This measurement time continued until the participants pushed the finish button. In parallel, the time when each positioning was finished was measured. This was used to clarify the number of points to be measured were the boundary to make a significant difference between the two systems.

The second criterion was the measurement accuracy of the distance between two points. The ground-truth distance was measured manually using a measuring tape. The absolute difference between the ground truth and the estimated result was the error.

To compare the performances of the two systems, we statistically analyzed the results of the operation time and the measurement accuracy by t-test, implemented on Microsoft Excel2016. A t-test can determine whether a significant difference exists between the averages of sample pairs. Additionally, outliers can be excluded using the interquartile range. To visualize the results, we used a box-and-whisker plot.

4.4 Result

4.4.1 Participant Statistics

Ten students from a university participated in the evaluation: 10 male; mean age, 22.6 years; age

range, 21 to 25; mean height, 170.5 cm. At the beginning of the evaluation, we used the pre-questionnaires of other evaluations (Polvi et al., 2016; Tashiro et al., 2019). The participants answered their experiences on a 7-point Likert scale (1 = not at all and 7 = very familiar). They estimated their experiences for touchscreen (M=6.6, SD=0.49), handheld devices (M=6.5, SD=0.50), AR (M=4.3, SD=1.27), handheld AR (M=4.2,SD=1.47), and 3D interface (M=3.8, SD=1.40). Compared with other evaluations (Polvi et al., 2016; Tashiro et al., 2019), the participants were familiar with not only the smartphone, but also AR and Handheld AR.

For the statistical analysis, 10 samples were used for the operation time of each method, as 1 sample was given by each participant. Also, 50 samples were used for the measurement accuracy, as 5 samples were given by each participant.

4.4.2 Operation Time

As illustrated in Figure 4, SnipAR required more time than PlanAR to complete all the tasks. For PlanAR, the participants spent the average of 83.4 s to create a plane at the beginning. In our evaluation setup, we selected a difficult situation such that it was not trivial to generate a plane on a rack board owing to occlusions and few feature points generated by ARCore. In other situations, less time is required if there were sufficient feature points that could facilitate plane generation. The average operation time required for one 3D positioning was 22.5 s and 7.6 s for SnipAR and PlanAR, respectively. This is because the number of operations for 3D positioning in PlanAR is less than that of SnipAR.

Next, we investigated the number of points to be measured on a plane were the boundary such that the distance between the two systems is statistically significant, as illustrated in Figure 5. When using the total operation time at the 8-th positioning, a significant difference was discovered thorough the t-test, as shown in Table 2. The label "df" means the degree of freedom. The p-value was computed with a two-sided test. A significant difference is implied if the p-value was less than 0.05. Hence, our results show that PlanAR is faster when approximately 8 target points are distributed on a plane. This is equivalent to measuring four distances of two points. Because the scene in our daily life is composed of many planes, measuring several distances on a plane is not a special situation. Therefore, our interactive plane-reconstruction-based approach can be useful in practice.

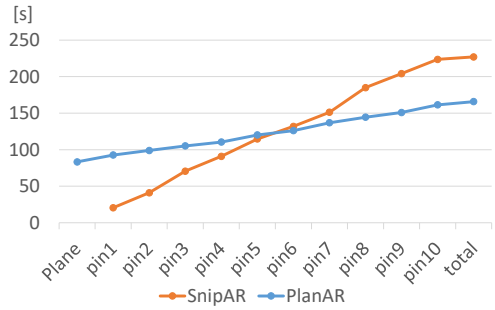


Figure 4: Operation times for PlanAR and SnipAR. The plane in the horizontal axis represents the time required for generating a plane. The digit in pin+digit represents the order for 3D positioning.

Table 2: Result of t-test for each box whisker.

The figure number	df	t-value	P-value
Figure 5(a)	9	0.927	0.378
Figure 5(b)	9	2.285	0.048
Figure 5(c)	9	3.302	0.009
Figure 6	45	1.753	0.086

4.4.3 Measurement Accuracy

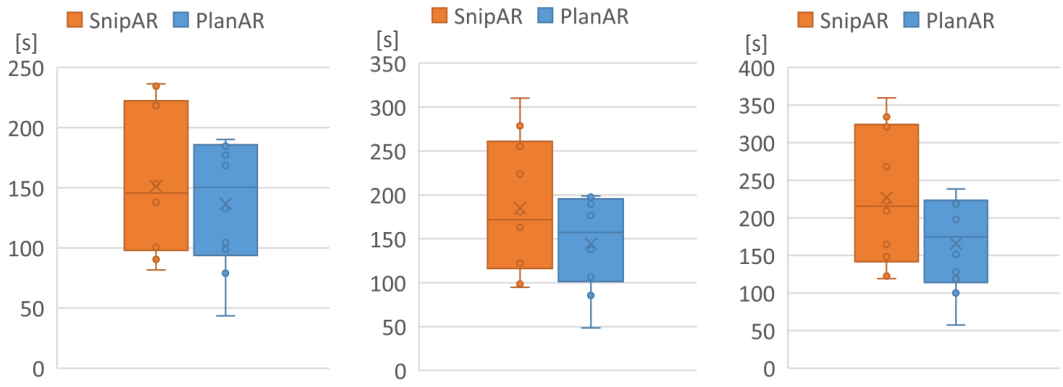
Figure 6 illustrates the result of the t-test for the measurement accuracy of distance between two points. The dots in the figure represent outliers detected using the interquartile range. The means of the error were 9.6 and 6.9 mm in SnipAR and PlanAR, respectively. A significant statistical difference was not observed in the measurement accuracy, as shown in table 2.

It is noteworthy that four obvious outliers appeared, in which the error was 10 times larger than those of others in the results of SnipAR. This occurs occasionally because of the instability of vSLAM in ARCore. Such obvious outliers were manually removed before the t-test; we used 46 samples in each method for the evaluation.

4.4.4 Number of Repeated Operations

In addition to the operation time and measurement accuracy, we investigated the number of repeated operations, which is related to the usability. The definition of a repeated operation in the evaluation is to repeat 3D positioning for the same position.

The average number of repeated operations was 0.4 and 2 times in SnipAR and PlanAR, respectively. This result is related to the required operations for each method. With SnipAR, the participants must select pixels at least twice at different camera views. To improve the accuracy, they are allowed to iterate the pixel selections at more views to adjust the estimated 3D position, if they desire. This means that the par-



(a) Time at 7-th positioning (b) Time at 8-th positioning (c) Time at the end of tasks

Figure 5: t-test for operation time at n-th positioning. At the 7-th positioning, a significant difference was not observed. After the 8-th positioning, a significant difference was observed. The results of t-test with these data are shown in Table 2.

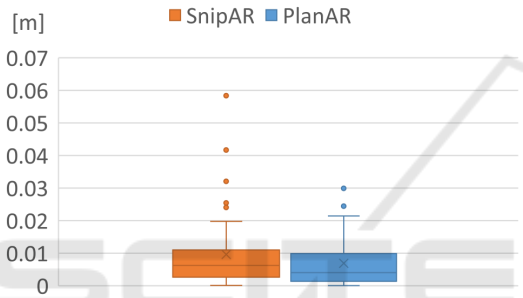


Figure 6: Measurement accuracy. Significant difference between the two methods was not observed.

ticipants tend to iterate rather than repeat pixel selections.

Conversely, the participants can perform 3D positioning with only one operation in PlanAR. If camera shake occurs when they are performing 3D positioning, it is hard to accurately select the target point. Therefore, the participants tend to repeat the operations. However, the operation time was not significantly affected by repeating operations because the operation for 3D positioning in PlanAR is simple.

4.4.5 Post-questionnaires

We followed the post-questionnaires used in other evaluations (Polvi et al., 2016; Tashiro et al., 2019). The results are summarized in Table 3. Overall, the scores of PlanAR were higher than those of SnipAR. In particular, PlanAR required less effort both physically and mentally than SnipAR. In terms of operation simplicity, the score of SnipAR was higher. This may be because PlanAR required the operation for creating planes before 3D positioning. Regarding screen flickering, the score of PlanAR was higher because the feature points were updated and displayed at every frame. Regarding information displayed on the screen, the results were almost the same. This is because the user interface for each system such as the button shape, color, number, and arrangement was unified maximally.

Considering the aforementioned results, SnipAR was useful in simple tasks such as 3D positioning for a few target points because of its simple manipulation. Meanwhile, PlanAR was suitable when the number of target points was increased because less time and effort required.

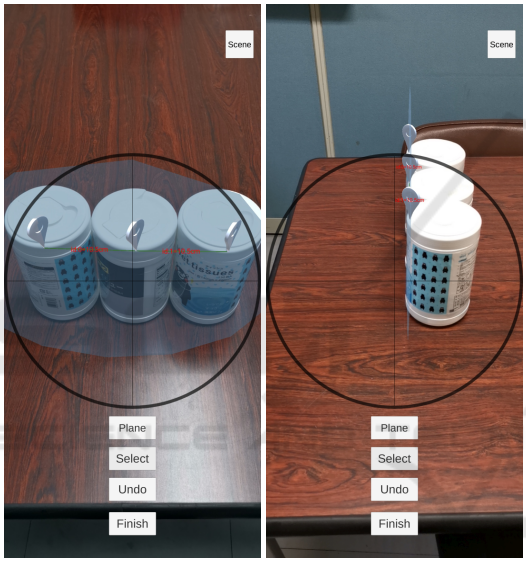
5 EXAMPLES OF PLANE RECONSTRUCTION

The significant property of our proposed system is that users can generate the geometric shape as they wish when map points from vSLAM are available. In other words, it is easy to generate a plane that is not easily reconstructed using automatic methods. In Figure 7, the tangent plane shared by several cylinders can be generated by selecting points on cylindrical bus bars. In Figure 8, the box shape can be computed from an opened box. In Figure 9, one plane can be generated between discontinuous planes by selecting points on both planes.

In our current system, only plane reconstruction is supported. It is easy to incorporate other geometric reconstruction methods for spheres, cylinders, and spline surfaces. The scene reconstruction with geometric shapes can suppress the error of noisy map points from vSLAM by averaging the error. Therefore, the accuracy of 3D positioning can be improved compared with meshes simply generated from map points. If the object shape in the scene is comprehen-

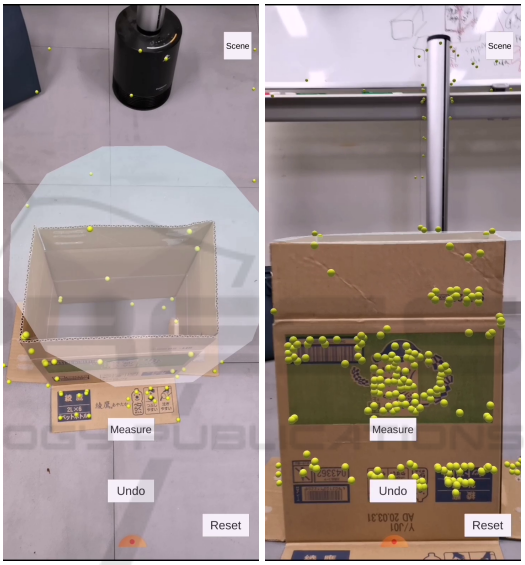
Table 3: Questionnaire and results (lower is better from 1 to 4, 7, and 8; higher is better from 5,6 and 9 to 11).

Question regarding manipulability	SnipAR	PlanAR
1. The interaction required significant body muscle effort	5.8	**4.9**
2. My arms and hands became tired	4.4	**2.9**
3. The device was difficult to hold	2.4	**2.1**
4. I lost grip or dropped the device	1.1	**1.0**
5. The manipulation was simple	**5.2**	4.7
6. The manipulation was easy to control	4.7	**4.8**
Question regarding Comprehensibility	SnipAR	PlanAR
7. The interaction required considerable mental effort	4.2	**3.0**
8. The display was not flickering considerably	**2.1**	2.9
9. The information displayed on the screen was sufficiently fast	5.9	**6.0**
10. The information displayed on the screen was appropriate	5.7	5.7
11. The information displayed on the screen was consistent	5.8	**5.9**



(a) Front view (b) Side view

Figure 7: Tangent plane reconstruction. Users select points on different cylinder surfaces to generate the tangent plane.



(a) Front view (b) Side view

Figure 8: Reconstruction from contour. Users select points on the contour.

sive for users, it would be extremely helpful for the system to obtain support from users to reconstruct the scene. If the user interaction is designed simply, the system usability can be improved with minimum user effort.

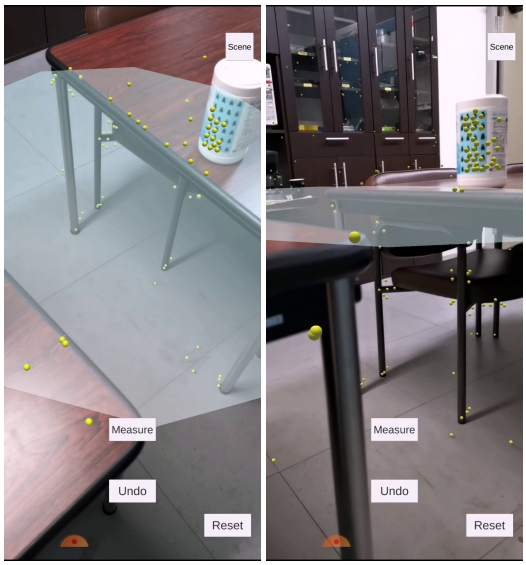
Furthermore, we attempted plane reconstruction when no plane existed. Under the circumstances, our method could generate planes in scenes that could not be detected by ARCore. Hence, this system can be applied widely.

6 CONCLUSIONS

We proposed a 3D positioning system with plane reconstruction and developed the method ”PlanAR” for this system, which was better than ”SnipAR” in terms of the measurement time when many annotations were placed. We herein focused on planes; however, this system can be utilized for other structures.

ACKNOWLEDGEMENTS

A part of this work was supported by JSPS KAKENHI, Grant Number JP18H04125 and JP18H04117.



(a) Front view (b) Side view

Figure 9: Reconstruction for discontinuous planes at the same height. Users select points on each plane to create a plane.

REFERENCES

Amann, M.-C., Bosch, T. M., Lescure, M., Myllylae, R. A., and Rioux, M. (2001). Laser ranging: a critical review of unusual techniques for distance measurement. *Optical engineering*, 40(1):10–20.

Baillot, Y., Brown, D., and Julier, S. (2001). Authoring of physical models using mobile computers. In *5th International Symposium on Wearable Computers*, pages 39–46. IEEE.

Delmerico, J. and Scaramuzza, D. (2018). A benchmark comparison of monocular visual-inertial odometry algorithms for flying robots. In *IEEE International Conference on Robotics and Automation*, pages 2502–2509. IEEE.

Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395.

Fuhrmann, S., Langguth, F., and Goesele, M. (2014). Mvea multi-view reconstruction environment. In *GCH*, pages 11–18.

Hartley, R. I. and Sturm, P. (1997). Triangulation. *Computer vision and image understanding*, 68(2):146–157.

Henrysson, A., Billinghurst, M., and Ollila, M. (2005). Virtual object manipulation using a mobile phone. In *International conference on Augmented tele-existence*, pages 164–171. ACM.

Hess-Flores, M., Recker, S., and Joy, K. I. (2014). Uncertainty, baseline, and noise analysis for l1 error-based multi-view triangulation. In *International Conference on Pattern Recognition*, pages 4074–4079. IEEE.

Hinckley, K., Pausch, R., Goble, J. C., and Kassell, N. F. (1994). A survey of design issues in spatial input. In *7th annual ACM symposium on User interface software and technology*, pages 213–222. ACM.

Jung, T., Gross, M. D., and Do, E. Y.-L. (2002). Annotating and sketching on 3d web models. In *7th International Conference on Intelligent User Interfaces*, IUI '02, pages 95–102, New York, NY, USA. ACM.

Lien, K.-C., Nuernberger, B., Höllerer, T., and Turk, M. (2016). Ppv: Pixel-point-volume segmentation for object referencing in collaborative augmented reality. In *IEEE International Symposium on Mixed and Augmented Reality*, pages 77–83. IEEE.

Marchand, E., Uchiyama, H., and Spindler, F. (2016). Pose estimation for augmented reality: a hands-on survey. *IEEE transactions on visualization and computer graphics*, 22(12):2633–2651.

Nuernberger, B., Lien, K.-C., Höllerer, T., and Turk, M. (2016). Interpreting 2d gesture annotations in 3d augmented reality. In *IEEE Symposium on 3D User Interfaces*, pages 149–158. IEEE.

Polvi, J., Taketomi, T., Yamamoto, G., Dey, A., Sandor, C., and Kato, H. (2016). Slidar: A 3d positioning method for slam-based handheld augmented reality. *Computers & Graphics*, 55:33–43.

Roberto, R. A., Uchiyama, H., Lima, J. P. S., Nagahara, H., Taniguchi, R.-i., and Teichrieb, V. (2017). Incremental structural modeling on sparse visual slam. *IPSJ Transactions on Computer Vision and Applications*, 9(1):5.

Rumpler, M., Irschara, A., and Bischof, H. (2011). Multi-view stereo: Redundancy benefits for 3d reconstruction. In *35th Workshop of the Austrian Association for Pattern Recognition*, volume 4, pages 1–8.

Schnabel, R., Wahl, R., and Klein, R. (2007). Efficient ransac for point-cloud shape detection. In *Computer graphics forum*, volume 26, pages 214–226. Wiley Online Library.

Taketomi, T., Uchiyama, H., and Ikeda, S. (2017). Visual slam algorithms: a survey from 2010 to 2016. *IPSJ Transactions on Computer Vision and Applications*, 9(1):16.

Tashiro, S., Uchiyama, H., Thomas, D., and Taniguchi, R.-i. (2019). 3d positioning system based on one-handed thumb interactions for 3d annotation placement. In *IEEE Conference on Virtual Reality and 3D User Interfaces*, pages 1181–1182. IEEE.

Whelan, T., Salas-Moreno, R. F., Glocker, B., Davison, A. J., and Leutenegger, S. (2016). Elasticfusion: Real-time dense slam and light source estimation. *The International Journal of Robotics Research*, 35(14):1697–1716.

Wither, J., DiVerdi, S., and Höllerer, T. (2009). Annotation in outdoor augmented reality. *Computers & Graphics*, 33(6):679–689.

Yu, F. and Gallup, D. (2014). 3d reconstruction from accidental motion. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3986–3993.

Zhang, Z. (1998). Determining the epipolar geometry and its uncertainty: A review. *International journal of computer vision*, 27(2):161–195.