Marker-less AR system based on line segment feature

Yusuke Nakayama[a], Hideo Saito[a], Masayoshi Shimizu[b] and Nobuyasu Yamaguchi[b]

[a]Keio University, 3-14-1 Hiyoshi, Kohoku-ku, Yokohama, Japan;
[b]Fujitsu Laboratories Ltd, 4-1-1 Kamikodanaka, Nakahara-ku, Kawasaki, Japan

ABSTRACT

We propose a method of marker-less AR which uses line segment feature. Estimating camera poses is important part of AR system. In most of conventional marker-less AR system, feature point matching between a model and its image is required for the camera pose estimation. However, a sufficient number of correspondence points is not always detected to estimate accurate camera poses. To solve this problem, we propose the use of line segments feature that can possibly be detected even when only a few feature points can be detected. In this paper, we propose a marker-less AR system that uses line segment features for camera pose estimation. In this system, we propose a novel descriptor of the line segment feature for achieving fast computation of the camera pose estimation. We also construct a database which contains a k-d tree of line feature and 3D line segment position for finding 2D-3D line segment correspondences from input image and the database, so that we can estimate the camera pose and perform AR. We demonstrated that our proposed method can estimate the camera pose and provide robust marker-less AR in the situation where point matching method fails.

Keywords: Line segment, The camera pose estimation, Marker-less AR, Line segment matching, Line segment feature

1. INTRODUCTION

Augmented Reality (AR) is a technology that enhances user's perception of the real world by embedding additional information in it. To overlay the additional information onto the real world, the camera pose must be estimated. Conventional methods for camera pose estimation can be categorized into two groups. One is based on tracking of markers and the other is marker-less tracking using scene features. The marker tracking system is called visual marker AR and it is robust for the camera pose estimation. Visual marker AR needs a marker. Therefore, users have to put a marker on the target scene where they want to perform AR. This is regarded as a troublesome task for users. On the other hand, since marker-less AR method enables us to estimate the camera pose without any marker, it is more desirable for AR. Roughly speaking, the camera pose estimation for maker-less AR is considered as estimating the rotation matrix and the translation vector from the matching between the 2D image and the 3D real world.

A lot of researches about marker-less AR have been done[1, 2] and most systems use feature points in the scene for obtaining the 2D-3D correspondences. The methods which use point features can estimate the camera pose robustly. However, in the scene where few feature points matching are detected between the 2D image and the 3D real world, the 2D-3D point correspondences cannot be obtained and estimation of the camera pose will fail. Especially in man-made situation there are many poorly textured objects. Feature points are rarely detected from texture less objects. Therefore, feature point based AR cannot deal with. In order to perform marker-less AR in this kind of situations, we need another scene feature instead.

Line segments which are in the scene can be considered as an alternative feature to solve this problem. A lot of line segments are detected in man-made situation even where few feature points are detected. There are several approaches[3, 4] for marker-less camera tracking based on line segments. However, these approaches get line

Further author information: (Send correspondence to Yusuke Nakayama)
Yusuke Nakayama: E-mail: nakayama@hvrl.ics.keio.ac.jp
Hideo Saito: E-mail: saito@hvrl.ics.keio.ac.jp
Masayoshi Shimizu: E-mail: shimizu.masa@jp.fujitsu.com
Nobuyasu Yamaguchi: E-mail: nobuyasu@jp.fujitsu.com

The Engineering Reality of Virtual Reality 2015, edited by Margaret Dolinsky, Ian E. McDowall,
Proc. of SPIE-IS&T Electronic Imaging, Vol. 9392, 93920I · © 2015 SPIE-IS&T
CCC code: 0277-786X/15/$18 · doi: 10.1117/12.2083673

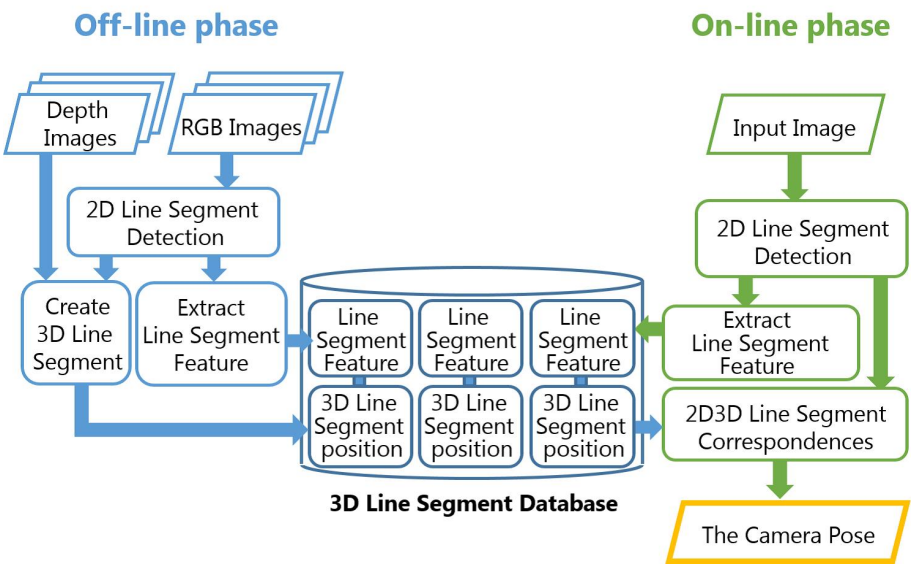


Figure 1. System Overview.

segment correspondences by the distance between the lines of the projected 3D model and image line. They do not use line segment feature descriptor for obtaining line correspondences. One of the reason that line segment feature descriptors are not used for AR methods is their computation time. Since the processing time is very important for AR, some line segment feature descriptors[5] which give high computation costs cannot be applied to AR.

In our previous work, we proposed a fast line segment feature descriptor which is called Directed LEHF.[6] Using this fast line segment descriptor, in this paper, we propose a marker-less AR system. This AR system is based on line segment feature. Therefore, we can perform AR where only a few feature points are detected. Moreover, using the fast line segment feature descriptor and forming k-d tree of the features provide less computation time for obtaining line segment correspondences.

Here is the overview of our proposed method. First of all, a 3D line segment database is constructed from a 3D line segment based model of the target scene. This database is generated from multiple images of the scene taken from RGB-D camera. It contains positions of 3D line segments and a k-d tree of Directed LEHF features from multiple angle. Next, 2D line segments from an input image are detected and Directed LEHF features are extracted. Nearest-neighbor matching between these input image ' s Directed LEHF features and the database is performed and the 2D-3D line segment correspondences are obtained. Finally, from these correspondences, the camera pose of the input image is estimated. Then, we can overlap CG onto the camera image using the estimated camera pose.

The experimental result shows that our proposed method using line segment matching can estimate the camera pose and perform AR even in the situation which has only a few feature points.

2. PROPOSED METHOD

In this section, we describe how to perform marker-less AR with line segments. Figure 1 shows an overview of camera pose estimation for AR. Our proposed method has off-line database generating phase and on-line camera tracking phase. In off-line phase, we capture the target scene by RGB-D camera. Then, we construct a database which contains positions of line segments in the 3D world coordinate and their projected 2D line segments' feature descriptor values. In on-line phase, 2D line segments from a camera frame are corresponded to the 3D line segments in the database with their descriptor values. Finally, this 2D-3D line segment correspondences provide the camera pose of input frame.

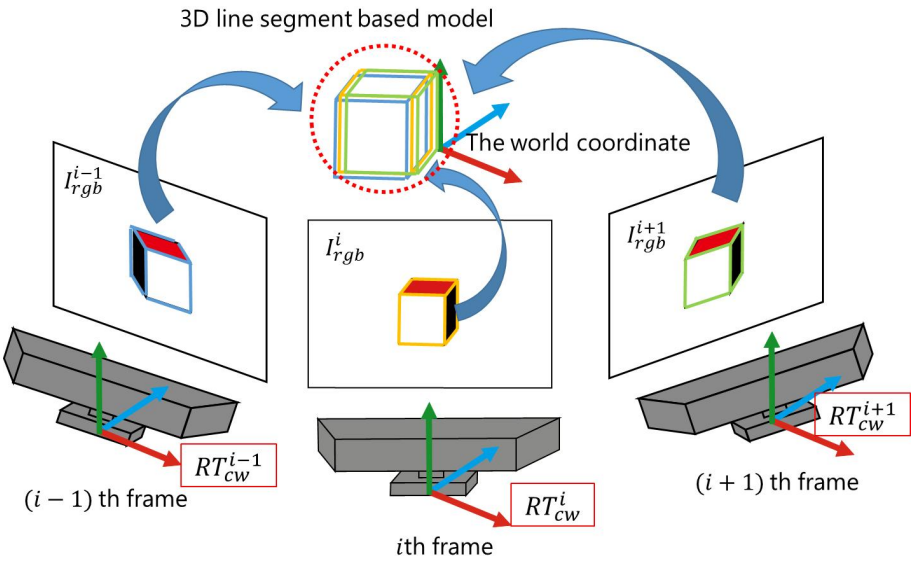


Figure 2. 3D line segment based model generation.

2.1 3D Line Segment Database Generation

This subsection describes how to generate the 3D line segment database. This database generation is operated in off-line and based on 3D line segment based model generation method which is our previous work.[7] This model generation method needs target scene's RGB images and Depth images captured by RGB-D camera as its inputs. Then, it provides 3D line segments in each frame's camera coordinate and each frame's relative camera pose to the first frame. Therefore, the 3D line segments in the camera coordinate are translated into the world coordinate by the relative camera poses. (This method assumes that the world coordinate is defined by the camera coordinate of the first frame.) Figure 2 shows this model generation. I_{rgb}^{i} denotes RGB image in ith frame and RT_{cw}^{i} denotes the relative camera pose of ith frame. From here, we explain about constructing 3D line segment database.

2.1.1 2D Line Segment Descriptor Value Extraction

As shown in Figure 2, each frame's 3D line segments are back-projected from 2D line segments detected in the RGB images. These 2D line segments are detected by a Line Segment Detector (LSD).[8] In the 2D line segments detection phase, 2D line segment feature descriptor values are also extracted. We used Directed LEHF as a line segment feature descriptor. Directed LEHF is an improve version of Line-based Eight-directional Histogram Feature (LEHF) which is a fast line segment descriptor proposed in Ref. 9. LEHF does not decide which edge point of line segment is start point or end point. Therefore, LEHF needs to compute descriptor distances from two directions, forward direction and inverse direction. On the other hand, Directed LEHF computes the distances one time because it decides a direction of line segment. In this time, we set Directed LEHF feature descriptor as 112 dimensional vector.

2.1.2 Construction of the 3D Line Segment Database

After extracting the 2D line segment features, these features of each frame and the back-projected 3D line segments' positions in the world coordinate are stored into the database. Due to the large number of line segment features, exhaustive search for closest matches might be extremely inefficient. Therefore, we employ a k-d tree search. A k-d tree is constructed from all line segment features. The 3D line segment database then has a k-d tree of 2D line segment features and each feature has reference to its 3D back-projected line segment's position.

2.2 On-line Camera Pose Estimation

With the 3D line segment database, the current camera pose is computed for live augmentation. In this subsection, the camera pose estimation is described. Suppose the camera intrinsic parameter has already been known. Then, what we will estimate is a transform matrix $RT_{cw} = [R|t]$ containing a 3×3 rotation matrix (R) and 3D translation vector (t).

First, 2D line segments are detected from the current camera frame, by employing LSD. Using LSD, we can obtain end points of each 2D line segment. Let these detected 2D line segments be $\mathcal{L} = \{l^0, l^1, \cdots, l^N\}$.

For each line segment in $\mathcal{L}$, Directed LEHF feature is extracted. To calculate the Directed LEHF feature descriptor, each line segment's start point and end point are decided by the method described in Ref. 6.

The features from the current frame are matched to the features in the 3D line segment database by the nearest neighbor search. In the database, each feature has its link to the 3D line segment. Therefore, 2D line segments from the current frame and 3D line segments from the database are corresponded by the Directed LEHF matching. Let the 3D line segments in the database be $\mathcal{L}_w = \{L_w^0, L_w^1, \cdots, L_w^M\}$. The set of 2D-3D line segment correspondence is represented as

$$\mathcal{LC} = \{(L_w^{g(j)}, l^{f(j)}), j = 0, 1, \cdots, K\}, \tag{1}$$

in which $(L_w^{g(j)}, l^{f(j)})$ represents a pair of 2D-3D line correspondences $g(j) \in [0, M]$ and $f(j) \in [0, N]$.

Given a set of 2D-3D line correspondences, we solve the Perspective n Lines (PnL) problem, then estimate the camera pose. (The PnL problem is a counterpart of the PnP problem for point correspondences.) However, $\mathcal{LC}$ may contain some mismatches. We use a method which solves the PnL problem with an algorithm like RANSAC,[10] and estimate the camera pose RT_{cw}. This method mainly use RPnL[11] for solving the PnL problem. Suppose we have $\mathcal{LC}$ which is K sets of 2D-3D line segment correspondences, we randomly select four 2D-3D line segment correspondences from $\mathcal{LC}$. Let the four set of 2D-3D line segment correspondences be represented as

$$\mathcal{LC}_{four} = \{(L_w^{a(k)}, l^{b(k)}), k = 0, 1, 2, 3\}, \tag{2}$$

in which $(L_w^{a(k)}, l^{b(k)})$ represents four pairs of 2D-3D line segment correspondences $a(k) \in [0, M]$ and $b(k) \in [0, N]$. Then, the rest of $(K - 4)$ 2D-3D line segment correspondences are represented as

$$\mathcal{LC}_{rest} = \{(L_w^{g(j)}, l^{f(j)}) | 0 \leq j \leq K, g(j) \neq a(k), f(j) \neq b(k), k = 0, 1, 2, 3\}. \tag{3}$$

With $\mathcal{LC}_{four}$, we solve the PnL problem using RPnL and estimate the camera pose RT'_{cw}. All of the 3D line segments L_w in $\mathcal{LC}_{rest}$ are projected to image frame by the camera intrinsic parameter and RT'_{cw}. Let the projected 2D line segments be l_w. Then we have pairs of the projected 2D line segments and the 2D line segments detected from the current frame. This set of pairs is represented as

$$\mathcal{LP}_{rest} = \{(l_w^{g(j)}, l^{f(j)}) | 0 \leq j \leq K, g(j) \neq a(k), f(j) \neq b(k), k = 0, 1, 2, 3\}. \tag{4}$$

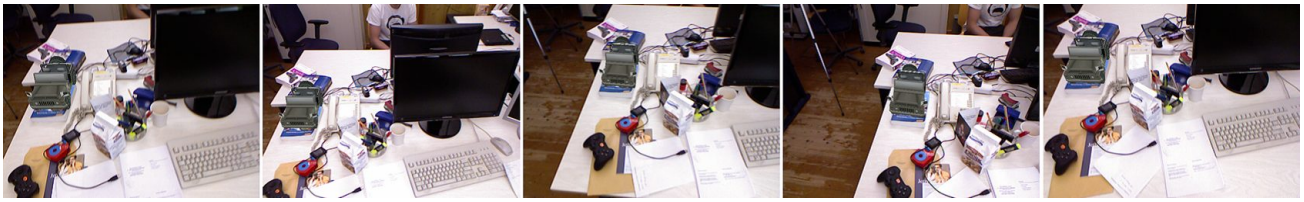
We calculate the error $e(j)$ between $l_w^{g(j)}$ and $l^{f(j)}$. We define $e(j)$ as

$$e(j) = S(j)/(length(l_w^{g(j)}) + length(l^{f(j)})), \tag{5}$$

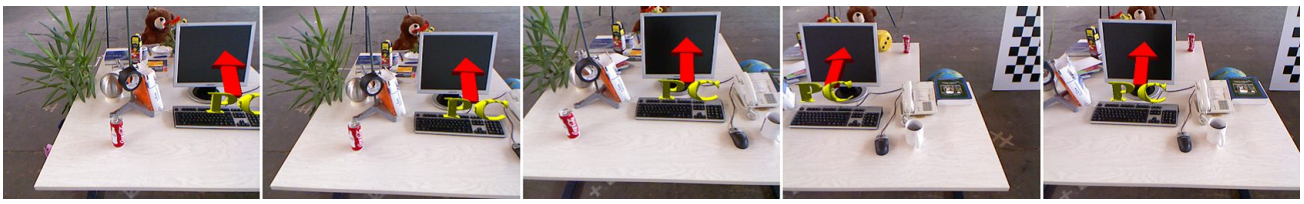
where $S(j)$ is an area of rectangle obtained by connecting four end points of $l_w^{g(j)}$ and $l^{f(j)}$, $length(l)$ is the length of 2D line segment l. The total of $e(j)$ is defined as an error given by RT'_{cw}.

We also randomly select another set of $\mathcal{LC}_{four}$ and repeat the steps explained above N_{RANSAC_INPUT} times to estimate RT'_{cw}. We choose RT'_{cw} which gives the smallest total of $e(j)$ as a tentative camera pose $tentativeRT_{cw}$. Next, using the camera intrinsic parameter and $tentativeRT_{cw}$, all of the 3D line segments $L_w^{g(j)}$ in $\mathcal{LC}$ are projected to the image frame and their projected 2D line segments $l_w^{g(j)}$ are obtained. We calculate $e(j)$ between $l_w^{g(j)}$ and $l^{f(j)}$ in $\mathcal{LC}$. If $e(j)$ is less than threshold (TH_INPUT_e), we save the 2D-3D line segment correspondences as inlier.

Finally, we compute the camera pose of the current frame using another algorithm for the PnL problem proposed by Kumar and Hanson.[12] This algorithm estimates the camera pose iteratively. It needs a set of 2D-3D line segment correspondences and initial camera pose as inputs. We take the inliers and $tentativeRT_{cw}$ as inputs, and obtain the camera pose the current frame RT_{cw} as output of the algorithm.



(a)



(b)

Figure 3. The result of Marker-less AR in the sequence of the TUM RGB-D benchmark, (a) freiburg1_xyz, (b) freiburg2_xyz.

3. EXPERIMENT

In this section, we present the result of experimental evaluations. We have performed our Marker-less AR system with several sequences. In each experiment, a 3D line segment based model is firstly generated. Then, input frame's camera poses are estimated with the generated model and CG models are augmented.

3.1 Accuracy of Estimated Camera Pose

This subsection describes the experiments for evaluating the error of the estimated camera pose. We tested our approach on two sequences of the TUM RGB-D benchmark[13] and a sequence in man-made situation.

3.1.1 TUM RGB-D Benchmark Sequences

Our proposed method needs RGB images and Depth images for the 3D line segment model generation. Therefore, we used two sequences of the TUM RGB-D benchmark for both off-line model generation phase and on-line camera pose estimation phase. In the on-line phase, we did not use depth images and used only RGB images. Then, the estimated camera poses are compared with the ground truth given from the TUM RGB-D benchmark. In this time, we set N_{RANSAC} to 1000 and TH_e to 0.01 for model generation (See Ref. 6 for details) and set N_{RANSAC_INPUT} to 50 and TH_INPUT_e to 5 for camera pose estimation. The used sequences of the TUM RGB-D benchmark are "freiburg1_xyz" and "freiburg2_xyz". In the experiment of "freiburg1_xyz", 50 frames are used for model generation and estimated 700 frames' camera poses. In the experiment of "freiburg2_xyz", 200 frames are used for model generation and estimated 700 frames' camera poses.

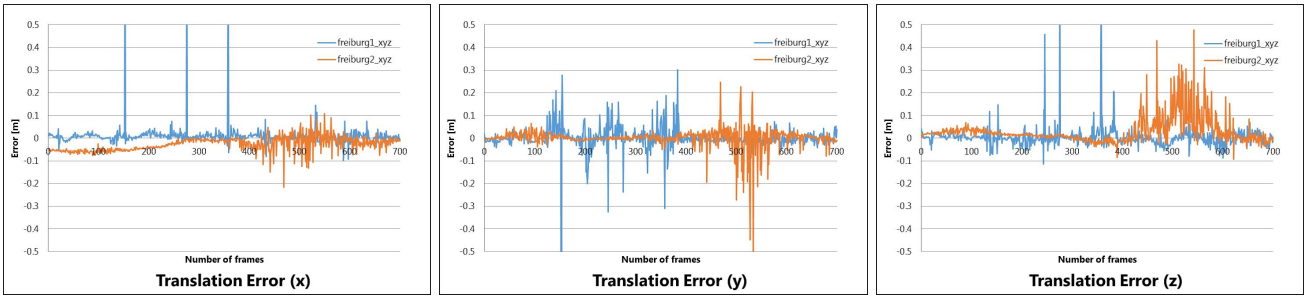
The snapshots from Figure 3 show the result of AR in the sequence of the TUM RGB-D benchmark. Figure 3(a) shows the scene of "freiburg1_xyz" that contains a typical office environment. We put the CG model of jeep onto the blue book in the environment. Figure 3(b) shows the sequence of "freiburg2_xyz". This sequence shows a desk scene. We put the CG instruction which indicates "PC" around the computer.

Figure 4 and Figure 5 shows the error of the estimated camera pose. As shown in Figure 4, almost all translation errors are below 0.5 m. Most frames' rotation errors are below 0.2 radians shown in Figure 5.

3.1.2 Man-made situation sequence

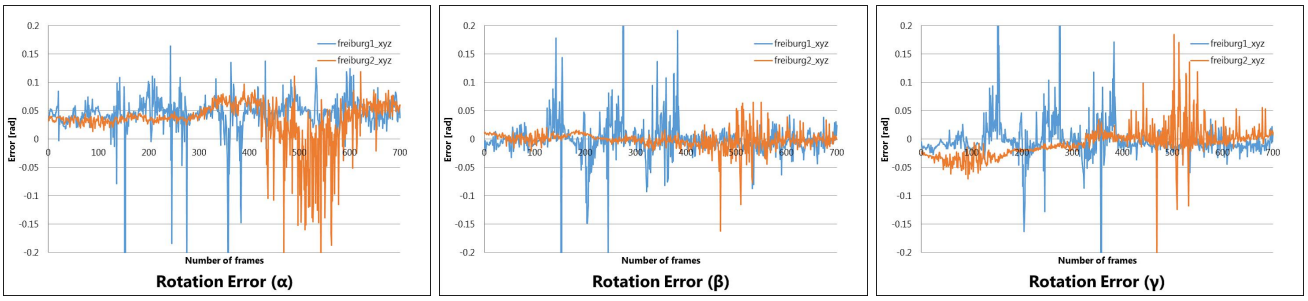
For demonstrating that our proposed method can estimate camera pose in man-made situation, we also tested it in another sequence. In this situation, mainly a simple door is shown. This environment contains only texture less objects. Therefore, some AR methods witch use feature points may fail.

In this time, we set N_{RANSAC} to 1000 and TH_e to 0.003 for model generation and set N_{RANSAC_INPUT} to 50 and TH_INPUT_e to 5 for camera pose estimation. 15 frames are used for model generation and we estimated 70 frames' camera poses.



(a) (b) (c)

Figure 4. Translation error.



(a) (b) (c)

Figure 5. Rotation error.

The results of our marker-less AR method in this situation are shown in Figure 6. We augmented the instruction of "Entering the Key number" at the door handle. Moreover we tested PTAM[1] which uses feature points for the camera tracking to the same situation and the results are shown in Figure 7.

The ground truth of this scene was not obtained. Then, we measured the re-projection errors using the projected four points shown in Figure 8 by the estimated camera pose for evaluating its accuracy. These re-projection errors are shown in Figure 9.

As shown in Figure 7, with PTAM, few feature points are detected and camera tracking was not accurate in this situation. However, as shown in Figure 6 and Figure 9, our proposed method can estimated the camera pose accurately even in few feature points situation.

Table 1. Average processing time of the individual steps

Individual step	time in ms
Loading frame	1.3
2D Line Segments Detection (LSD)	47.7
Directed LEHF Feature Extraction	5.1
2D-3D Line Segment Correspondences	3.6
Solving the PnL problem	77.4
Draw the CG model by OpenGL	11.1
Total	149.7

3.2 Processing time

Our proposed method is Marker-less AR system. Therefore, processing time is important. We measured the processing time of the online phase with the same dataset used in 3.1.2 (The door scene). An analysis of the processing time is carried out on a Intel(R) Core(TM) i7-3520M CPU with 2.90GHz. The average computational costs for every individual step are shown in Table.1. This result is average of 70 frames. As shown in Table.1, this system run for this particular example with a frame rate about 6.7fps.

Figure 6. The result of Marker-less AR in the sequence of man-made situation.

Figure 7. The result of PTAM in the sequence of man-made situation.

4. CONCLUSION

In this paper, we propose model based marker-less AR system which uses line segment feature. Most of marker-less AR method make use of feature points for obtaining correspondences between real world and camera frame. These feature points method cannot deal with some situation that contains texture-less objects. Even in the scene where only a few feature points are detected, line segments are detected a lot. Then, line segment feature can be an alternative feature. However, simply using line segment feature descriptor for obtaining 2D-3D correspondences takes quite long time. Therefore, we performed real-time marker-less AR system using line segment feature with a fast line segment descriptor and creation of its k-d tree. In our proposed method, the target scene's 3D line segment based model is firstly generated. Then 3D line segments' position and their projected line segments' Directed LEHF descriptor values are stored into the 3D line segment database. In this time, Directed LEHF values are stored as k-d tree for reducing the searching time. In on-line phase, current frame's 2D line segments are detected and Directed LEHF values are extracted. Nearest-neighbor matching between these Directed LEHF features and the database is performed for obtaining the 2D-3D line segment correspondences. Lastly, the correspondences provide camera pose of current frame. In the experiments, we demonstrated that our marker-less AR system can be performed in some sequences. Moreover, our approach using line segments gave accurate camera tracking even in the situation where feature points method cannot deal with.

ACKNOWLEDGMENTS

This work was partially supported by MEXT/JSPS Grant-in-Aid for Scientific Research(S) 24220004, and JST CREST "Intelligent Information Processing Systems Creating Co-Experience Knowledge and Wisdom with Human-Machine Harmonious Collaboration".

REFERENCES

[1] Klein, G. and Murray, D., "Parallel tracking and mapping for small AR workspaces," in [*6th IEEE and ACM International Symposium on Mixed and Augmented Reality*], 225–234, IEEE (2007).

[2] Skrypnyk, I. and Lowe, D. G., "Scene modelling, recognition and tracking with invariant image features," in [*Third IEEE and ACM International Symposium on Mixed and Augmented Reality*], 110–119, IEEE (2004).

[3] Wuest, H., Vial, F., and Strieker, D., "Adaptive Line Tracking with Multiple Hypotheses for Augmented Reality," in [*Fourth IEEE and ACM International Symposium on Mixed and Augmented Reality*], 62–69, IEEE (2005).

Figure 8. The four points used for measuring re-projection errors.

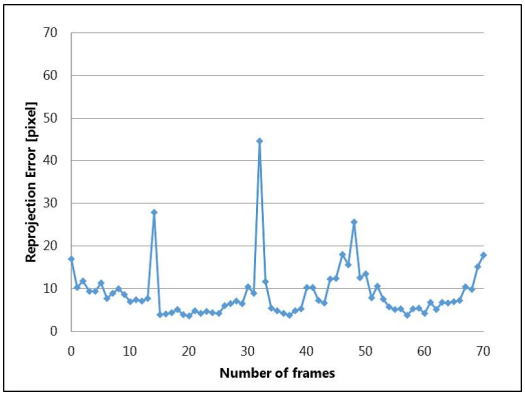


Figure 9. Re-projection errors.

[4] Wuest, H., Wientapper, F., and Stricker, D., "Adaptable Model-Based Tracking Using Analysis-by-Synthesis Techniques," in [*Computer analysis of images and patterns*], 20–27, Springer (2007).

[5] Wang, Z., Wu, F., and Hu, Z., "MSLD: A robust descriptor for line matching," *Pattern Recognition* **42**(5), 941–953 (2009).

[6] Nakayama, Y., Honda, T., Saito, H., Shimizu, M., and Yamaguchi, N., "Accurate Camera Pose Estimation for KinectFusion Based on Line Segment Matching by LEHF," in [*Proceedings of the International Conference on Pattern Recognition*], 2149–2154 (2014).

[7] Nakayama, Y., Saito, H., Shimizu, M., and Yamaguchi, N., "3D Line Segment Based Model Generation by RGB-D Camera for Camera Pose Estimation," in [*3rd International Workshop on Intelligent Mobile and Egocentric Vision*], (2014).

[8] Von Gioi, R. G., Jakubowicz, J., Morel, J.-M., and Randall, G., "LSD: A fast line segment detector with a false detection control," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **32**(4), 722–732 (2010).

[9] Hirose, K. and Saito, H., "Fast Line Description for Line-based SLAM," in [*Proceedings of the British Machine Vision Conference*], 83.1–83.11 (2012).

[10] Fischler, M. A. and Bolles, R. C., "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," *Communications of the ACM* **24**(6), 381–395 (1981).

[11] Zhang, L., Xu, C., Lee, K.-M., and Koch, R., "Robust and Efficient Pose Estimation from Line Correspondences," in [*Computer Vision – ACCV 2012*], 217–230, Springer (2013).

[12] Kumar, R. and Hanson, A. R., "Robust Methods for Estimating Pose and a Sensitivity Analysis," *CVGIP: Image Understanding* **60**(3), 313–342 (1994).

[13] Sturm, J., Engelhard, N., Endres, F., Burgard, W., and Cremers, D., "A Benchmark for the Evaluation of RGB-D SLAM Systems," in [*Proc. of the International Conference on Intelligent Robot Systems (IROS)*], (Oct. 2012).