3D Line Segment Based Model Generation by RGB-D Camera for Camera Pose Estimation

Yusuke Nakayama†, Hideo Saito†,
Masayoshi Shimizu‡, and Nobuyasu Yamaguchi‡

†Graduate School of Science and Technology, Keio University
3-14-1 Hiyoshi, Kohoku-ku, Yokohama, Japan
‡Fujitsu Laboratories Ltd
4-1-1 Kamikodanaka, Nakahara-ku, Kawasaki, Japan

Abstract. In this paper, we propose a novel method for generating 3D line segment based model from an image sequence taken with a RGB-D camera. Constructing 3D geometrical representation by 3D model is essential for model based camera pose estimation that can be performed by corresponding 2D features in images with 3D features of the captured scene. While point features are mostly used as such features for conventional camera pose estimation, we aim to use line segment features for improving the performance of the camera pose estimation. In this method, using RGB images and depth images of two continuous frames, 2D line segments from the current frame and 3D line segments from the previous frame are corresponded. The 2D-3D line segment correspondences provide camera pose of the current frame. All of 2D line segments are finally back-projected to the world coordinate based on the estimated camera pose for generating 3D line segment based model of the target scene. In experiments, we confirmed that the proposed method can successfully generate line segment based models, while 3D models based on the point features often fail to successfully represent the target scene.

1 Introduction

Generating 3D geometrical models of the environment is an essential technology for a lot of vision-based applications. For example, it is almost impossible to estimate camera pose for AR applications from the captured image sequence without such geometrical 3D model of the object environment. For constructing 3D models, Structure-from-Motion (SfM) approaches are often used. Traditional SfM approaches like [1] take correspondences from images, then recover 3D geometrical model of the scene and pose of the camera at each frame. To obtain these correspondences, feature points are generally detected, and then the detected features are matched between different frames. For the matching, feature point descriptors such as the scale-invariant feature transform (SIFT)[2] and speeded-up robust features (SURF)[3] are often used. However, in the situation where only a few feature points are detected, feature point matching may fail, therefore reconstructed objects are inaccurate. In man-made environment, there are usually a lot of untextured objects which do not provide sufficient number of feature points.

Line segment features can be considered as an alternative feature to solve this problem. A lot of line segments are detected in a man-made situation even where only a few feature points are detected. Over the years, several methods about line based 3D reconstruction have been reported. Comparing with feature point based reconstruction, the number of researches about line based is very few. This is mainly because line segment matching is still challenging task. Line segments have less distinctive appearance so that a descriptor of line segment feature is difficult to be defined. Therefore, some existing methods[4–6] do not rely on appearance-based line segment matching. However, recently, researches about line segment feature descriptor such as the mean standard-deviation line descriptor (MSLD)[7] or the Line-based Eight-directional Histogram Feature(LEHF)[8] have been reported. With these line segment feature descriptors, line segment based 3D model can be generated in man-made environment.

As one more advantage of line segments, line segment matching can be robust to large viewpoint changes. This is because line segments provide more information from their length, geometry characteristics and relative positions. One of the applications for generated 3D models is the estimation of camera pose using 3D models. With 3D models which consists feature points, the camera pose estimation cannot deal with large viewpoint changes because feature point matching is not robust to changes in perspective. On the other hand, thanks to the line segment's advantage, 3D line segment based model and estimation of the camera pose from it can deal with the perspective changes. Therefore, generating 3D line segment based model is also important for the large viewpoint change situation.

In this paper, we propose a novel method for generating 3D line segment based model from an image sequence taken with a RGB-D camera. For obtaining line segments matching between images, we use Directed LEHF[9] which is an improved version of LEHF[8]. In the proposed method, 2D line segments are first detected from RGB images. Then, 2D line segments between consecutive two frames are matched by Directed LEHF. From this matching, the 2D line segments of the previous frame can be transferred to 3D line segments in the world coordinate. Then, the current frame's 2D line segments and the previous frame's 3D line segments can be corresponded. These 2D-3D correspondences give camera pose of the current frame by solving the Perspective-n-Lines (PnL) problem. With this estimated camera pose, the 2D line segments of current frame are translated to 3D line segments. This procedure is repeated and objects represented by 3D line segments are obtained as a 3D model. We have experimentally demonstrated the performance of the proposed method by comparison with other 3D reconstruction methods. The experimental result shows that our proposed method using line segment matching can generate an accurate 3D model in the situation which has few feature points. Moreover, we demonstrated that our 3D line segment based model can use for estimating camera poses in the large viewpoint changes situation.

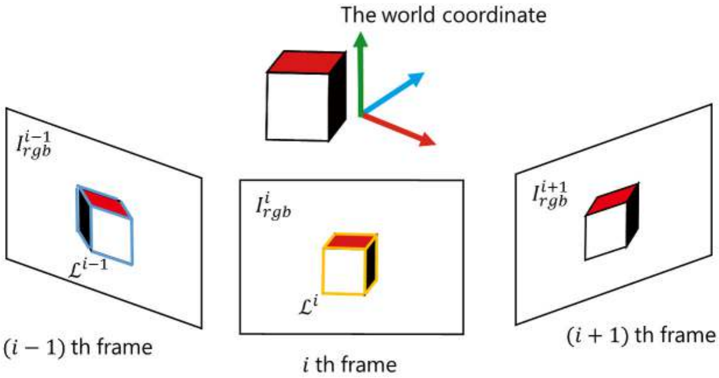


Fig. 1. Detection of 2D line segments. The blue line segments from I_{rgb}^{i-1} and the yellow line segments from I_{rgb}^{i} are $\mathcal{L}^{i-1}$ and $\mathcal{L}^{i}$, respectively.

2 Proposed Method

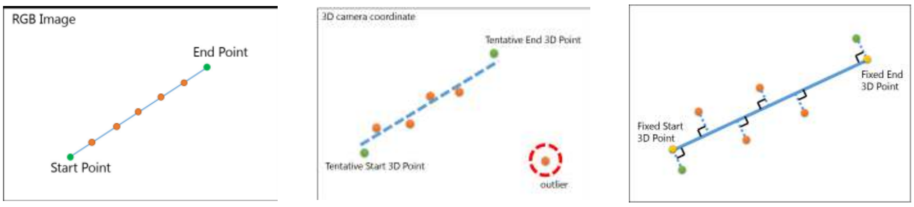
In this section, we describe the proposed method for generating a 3D line segment based model of a target scene from an image sequence captured by moving a RGB-D camera, such as Kinect. N multiple images of a target scene are captured by RGB-D camera. Suppose we have N RGB images $\{I_{rgb}^{N}\}$ and N depth images $\{I_{d}^{N}\}$ captured with the Kinect. The basic idea is that each frame's camera pose can be estimated from its previous frame's camera pose, and then 2D line segments on RGB Images are back-projected into the 3D world coordinate by the estimated camera pose. The camera pose at ith frame is represented a transform matrix $RT_{cw}^{i} = [R_i|t_i]$ containing a 3×3 rotation matrix (R_i) and 3D translation vector (t_i). In the following subsections, we describe 3D geometry of the consecutive two frames, $(i-1)$th and ith frame. Then we will describe the way of computing RT_{cw}^{i} using known $(i-1)$th frame's camera pose RT_{cw}^{i-1}. We set the first frame's camera pose RT_{cw}^{0} as 4×4 Identity matrix, because we assume that the world coordinate is defined by the camera coordinate of the first frame.

2.1 Detection of 2D line segments

First, 2D line segments are detected from RGB images, by employing the fast line segment detector (LSD)[10]. Using LSD, we can obtain end points of each 2D line segment. As shown in Fig. 1, two sets of line segments detected from I_{rgb}^{i-1} and I_{rgb}^{i} by LSD are indicated by $\mathcal{L}^{i-1} = \{l_{i-1}^{0}, l_{i-1}^{1}, \cdots, l_{i-1}^{M_{i-1}}\}$ and $\mathcal{L}^{i} = \{l_{i}^{0}, l_{i}^{1}, \cdots, l_{i}^{M_i}\}$, respectively.

2.2 Creation of 3D line segments

We also have depth images, I_{d}^{i-1} and I_{d}^{i}. Therefore, the positions of the 3D line segments in the camera coordinate of $(i-1)$th frame and ith frame are



(a) (b) (c)

Fig. 2. The way to create 3D line segments. (a) 2D points on the line segment, (b) 3D points in the camera coordinate, (c) the 3D line segment and fixed end points.

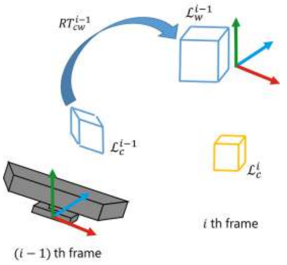


Fig. 3. Creation of the 3D line segments.

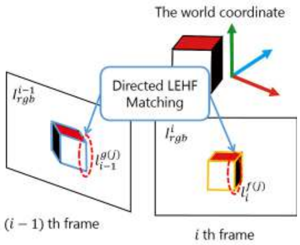


Fig. 4. Directed LEHF matching.

obtained, so that we can create 3D line segments in the camera coordinate. Suppose we choose one 2D line segment in $\mathcal{L}^{i-1}$ and $\mathcal{L}^{i}$. First, as shown in Fig. 2(a), we get equally-spaced points on the line segment between the start point and the end point. These acquired points, the start point and the end point are translated from the image coordinate to the camera coordinate using depth value. These translated 3D points are supposed to be on the same 3D line. However, because of the Kinect ' s depth value error, the 3D points also have error, therefore, they are not on the same 3D line. To decrease the error, RANSAC[11] is used to find inliers of the 3D points. Thus, we can eliminate points which stay from the others. In this process, as shown in Fig. 2(b), the two points which are located at the either end of the inliers are assumed as a tentative start 3D point and end 3D point. Next, we compute a 3D line which minimizes the lengths of perpendiculars from each inlier point to the 3D line. Finally, we chose the extremities of the perpendiculars from the tentative start point and end point as a fixed start 3D point and a fixed end 3D point. Therefore, connecting these fixed start 3D point and end 3D point, we obtain a 3D line segment. Fig. 2(c) shows the fixed two points and the obtained 3D line segment. By applying this procedure to every line segment in $\mathcal{L}^{i-1}$ and $\mathcal{L}^{i}$, we obtain $\mathcal{L}_c^{i-1} = \{L_{c,i-1}^0, L_{c,i-1}^1, \cdots, L_{c,i-1}^{M_{i-1}}\}$ and $\mathcal{L}_c^{i} = \{L_{c,i}^0, L_{c,i}^1, \cdots, L_{c,i}^{M_i}\}$, where $L_{c,i}$ is translated from l_i into the camera coordinate. These 3D line segments in $\mathcal{L}_c^{i-1}$ is translated from the camera coordinate to the world coordinate by known RT_{cw}^{i-1}. Therefore, we get $\mathcal{L}_w^{i-1} = \{L_{w,i-1}^0, L_{w,i-1}^1, \cdots, L_{w,i-1}^{M_{i-1}}\}$, where $L_{w,i}$ is translated from l_i into the world coordinate, as shown in Fig. 3.

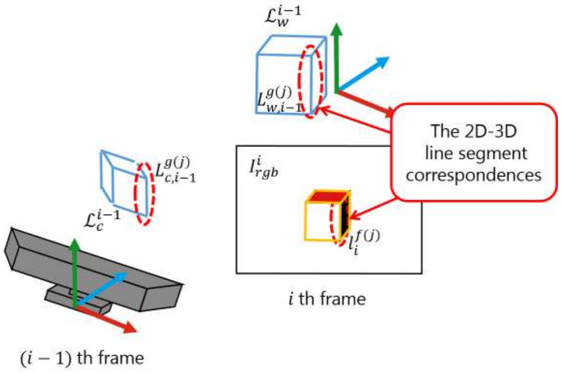


Fig. 5. The 2D-3D line segment correspondences. $l_{i-1}^{g(j)}$ from $\mathcal{L}^{i-1}$ and $l_i^{f(j)}$ from $\mathcal{L}^i$ are matched. $l_{i-1}^{g(j)}$ is transfered to $L_{c,i-1}^{g(j)}$ in $\mathcal{L}_c^{i-1}$. Then $L_{c,i-1}^{g(j)}$ is translated to $L_{w,i-1}^{g(j)}$ in $\mathcal{L}_w^{i-1}$. Therefore $L_{w,i-1}^{g(j)}$ and $l_i^{f(j)}$ are corresponded.

2.3 Matching 2D line segments by Directed LEHF

Next, we obtain 2D line segment matching between two images. To evaluate similarity of 2D line segments, we use Line-based Eight-directional Histogram Feature (LEHF)[8]. LEHF is a descriptor of line segments based on the gradient histogram for eight different directions along with the line segment. LEHF descriptor cannot deal with rotation change more than 180 degree. To get matching of the opposite direction line segments, LEHF needs to compute descriptor distances from two directions, forward direction and inverse direction. However, if the descriptor distances from the two directions are similar, mismatching is occurred. To improve the matching performance, we proposed Directed LEHF [9]. We adopt Directed LEHF as a line segment feature descriptor.

Using a method explained in [12], the Directed LEHF defines the direction of each 2D line segment by the average intensity of the gradient of the perpendicular direction with the line segment at all the points on the line segment. The sign of the average determines the direction of the line segment. The direction of the line segment determines the start point and end point of each 2D line segment in $\mathcal{L}^{i-1}$ and $\mathcal{L}^i$.

We search for 2D line segment matchings with $\mathcal{L}^{i-1}$ and $\mathcal{L}^i$ by the Directed LEHF matching. The resulting set of matching 2D line segment is represented as

$$\mathcal{LM}^i = \{(l_{i-1}^{g(j)}, l_i^{f(j)}), j = 0, 1, \cdots, K^i\}, \tag{1}$$

in which $(l_{i-1}^{g(j)}, l_i^{f(j)})$ represents a pair of matching 2D line segments $g(j) \in [0, M_{i-1}]$ and, $f(j) \in [0, M_i]$, as shown in Fig. 4.

2.4 2D-3D line segment correspondences

In the 2D line segment matching $\mathcal{LM}^i$, $l_{i-1}^{g(j)}$ from $\mathcal{L}^{i-1}$ and $l_i^{f(j)}$ from $\mathcal{L}^i$ are matched. The 3D line segment back-projected from $l_{i-1}^{g(j)}$ into the world coor-

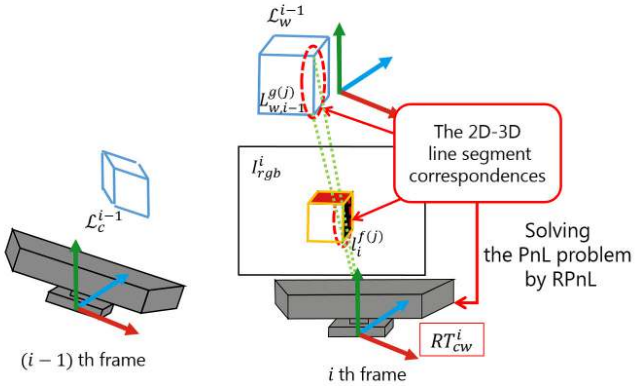


Fig. 6. Camera pose estimation by solving the PnL problem.

dinate is $L_{w,i-1}^{g(j)}$ in $\mathcal{L}_w^{i-1}$. Then, the 3D line segment in $\mathcal{L}_w^{i-1}$ and the 2D line segment in $\mathcal{L}^i$ are brought to be correspondence with the matching of $\mathcal{LM}^i$. The set of 2D-3D line segment correspondences is represented as

$$\mathcal{LC}^i = \{(L_{w,i-1}^{g(j)}, l_i^{f(j)}), j = 0, 1, \cdots, K^i\}, \tag{2}$$

in which $(L_{w,i-1}^{g(j)}, l_i^{f(j)})$ represents a pair of 2D-3D line correspondences $g(j) \in [0, M_{i-1}]$ and $f(j) \in [0, M_i]$. The position of the 2D line segment and the 3D line segment is shown in Fig. 5.

2.5 Solution for the PnL problem

Given a set of 2D-3D line correspondences, we solve the PnL problem, then estimate the camera pose. (The PnL problem is a counterpart of the PnP problem for point correspondences.) However, there is a fear that $\mathcal{LC}^i$ contains some mismatches. We use a method which solves the PnL problem with an algorithm like RANSAC explained in [9], and estimate the camera pose RT_{cw}^i. This method mainly use RPnL[13] for solving the PnL problem. Suppose we have $\mathcal{LC}^i$ which is K^i sets of 2D-3D line segment correspondences, we randomly select four 2D-3D line segment correspondences from $\mathcal{LC}^i$. This is because the program of RPnL needs at least four correspondences. Let the four set of 2D-3D line segment correspondences be represented as

$$\mathcal{LC}_{four}^i = \{(L_{w,i-1}^{a(k)}, l_i^{b(k)}), k = 0, 1, 2, 3\}, \tag{3}$$

in which $(L_{w,i-1}^{a(k)}, l_i^{b(k)})$ represents four pairs of 2D-3D line segment correspondences $a(k) \in [0, M_{i-1}]$ and $b(k) \in [0, M_i]$. Then, the rest of $(K^i - 4)$ 2D-3D line segment correspondences are represented as

$$\mathcal{LC}_{rest}^i = \{(L_{w,i-1}^{g(j)}, l_i^{f(j)}) | 0 \leq j \leq K^i, g(j) \neq a(k), f(j) \neq b(k), k = 0, 1, 2, 3\}. \tag{4}$$

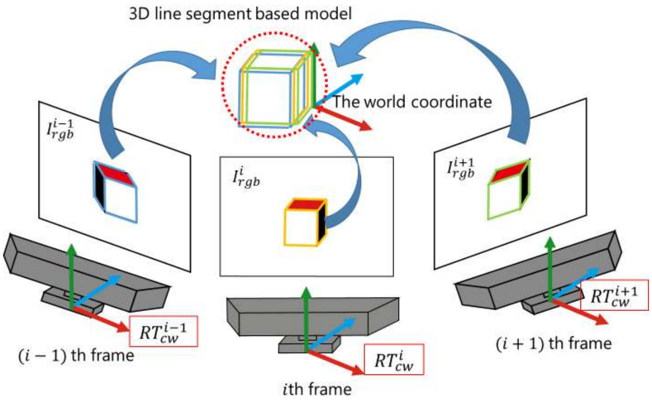


Fig. 7. The 3D line segment based model.

With $\mathcal{LC}^i_{four}$, we solve the PnL problem using RPnL and estimate the camera pose RT'_{cw}. The 3D line segments in $\mathcal{L}^i_c$ which are back-projected from $l^{f(j)}_i$ in $\mathcal{LC}^i_{rest}$ are translated from the camera coordinate into the world coordinate by RT'_{cw}. Let the 3D line segment in the world coordinate translated by RT'_{cw} be $L^{'f(j)}_{w,i}$. We calculate the error $e(j)$ between $L^{g(j)}_{w,i-1}$ from $\mathcal{LC}^i_{rest}$ and $L^{'f(j)}_{w,i}$. We define $e(j)$ as

$$e(j) = S(j)/(length_{w,i-1} + length_i), \tag{5}$$

where $S(j)$ is an area of rectangle obtained by connecting four end points of $L^{g(j)}_{w,i-1}$ and $L^{'f(j)}_{w,i}$, $lenght_{w,i-1}$ is length of $L^{g(j)}_{w,i-1}$, and $length_i$ is length of $L^{'f(j)}_{w,i}$. The total of $e(j)$ is defined as error given by RT'_{cw}.

We also randomly select another set of $\mathcal{LC}^i_{four}$ and repeat the steps explained above N_{RANSAC} 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 $tentativeRT_{cw}$, all of the 3D line segments in $\mathcal{L}^i_c$ which are back-projected from $l^{f(j)}_i$ are translated to the 3D line segments in the world coordinate $L^{'f(j)}_{w,i}$. We calculate $e(j)$ and if $e(j)$ is less than threshold (TH_e), we save the 2D-3D line segment correspondences as inlier.

Finally, we compute the camera pose of ith frame using another algorithm for the PnL problem proposed by Kumar and Hanson[14]. 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 of ith frame RT^i_{cw} as output of the algorithm. Fig. 6 shows this procedure.

Once RT^i_{cw} is obtained, all of the 3D line segments in the i th camera coordinate $\mathcal{L}^i_c$ are translated to the 3D line segments in the world coordinate $\mathcal{L}^i_w$. These procedures discussed above are repeated in every consecutive two frame. Then, each frame's camera pose and 3D line segments in the world coordinate are obtained. The 3D line segments of every frame are 3D line segment based

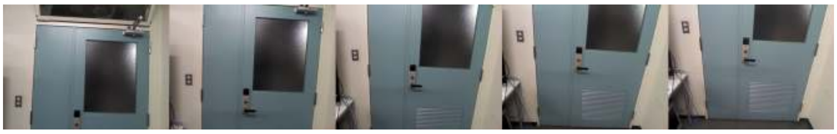


Fig. 8. Parts of the input RGB images.

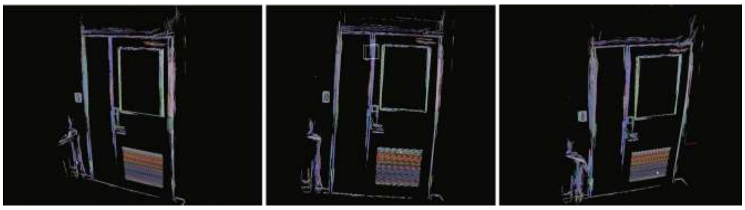


Fig. 9. The 3D line segment based model of proposed method.

model as an output of our proposed method. Fig. 7 shows the concept of the 3D line segment based model generation.

3 Experiment

We conducted two experiments for generating 3D models. One is to show that our proposed method can generate a 3D model of the scene where other methods cannot generate accurate models. Another is for demonstrating that camera pose estimation by our proposed model can perform well in large viewpoint changes. In both experiments, we compared our proposed method with a method which uses SIFT feature point matching instead of Directed LEHF matching. For model generation by our method, we set N_{RANSAC} to 5000 and TH_e to 0.003.

3.1 Experiment 1 : Generating 3D model

First of all, we generated 3D model of a scene by our proposed method. We used 81 frames as input image sequence. Fig. 8 shows some of the input RGB images. Fig. 9 shows the 3D line segment based model of the scene as an output of our proposed method. Each frame's camera pose was estimated and 2D line segments detected in RGB images are back-projected into the 3D coordinate. With the estimated camera poses and each frame's depth image, we also back-projected every points from RGB images into the world coordinate. Then we obtain reconstructed object shape of the scene represented by colored point clouds. Fig. 10 shows virtual viewpoint images of the point clouds. In Fig. 9, line segments on the same physical edges of the door structure detected in the different frames are almost overlapping in the 3D space. This demonstrates that the camera poses of the image sequence are correctly estimated by our method.

Fig. 10 also demonstrates the same fact, because the colors rendered by back projection from different frames are almost matching so that the virtual view images are synthesized without any blur.

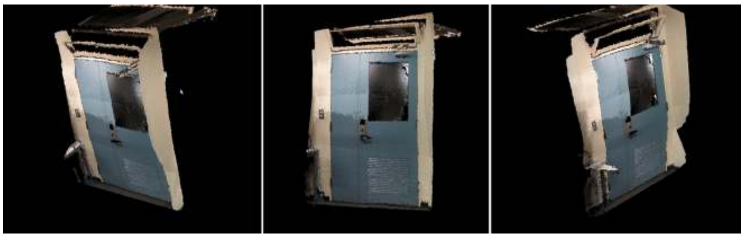


Fig. 10. The reconstructed object shape represented by point clouds.

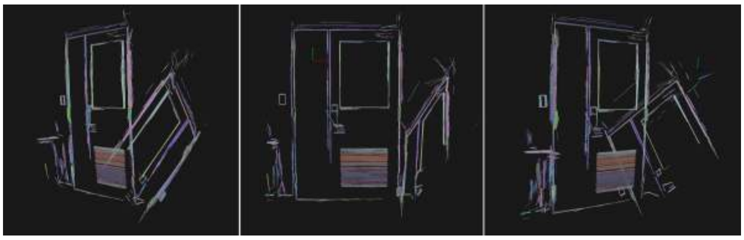


Fig. 11. The 3D line segment based model using SIFT feature point matching.

Fig. 9 and Fig. 10 demonstrate that our proposed method could estimate accurate camera pose of each frame and generate 3D models.

For comparison purpose, we conducted the same experiment with SIFT feature point matching instead of Directed LEHF matching for estimating camera pose of every frame. After camera pose estimation, the same procedure of generating the 3D line segment based model is also performed for detected line segments by LSD, which is also the same as our proposed method. Fig. 11 shows the 3D line segment based model using the camera pose estimated by SIFT feature point matching. Compared with Fig. 9 and Fig. 10, camera poses estimated by the method with feature point matching are less accurate than proposed method. This is because the scene does not provide sufficient number of SIFT feature points.

Next, we tried to get a 3D model of the same scene by other methods. Fig. 12 shows the 3D model reconstructed by Autodesk 123D Catch[15]. Autodesk 123D Catch is a free web service which reconstructs 3D object from images. We used 70 images which are included in the 81 images used in previous model generation. As shown in Fig. 12, the reconstructed model is distorted and inaccurate. Autodesk 123D Catch needs images which are captured from various angles. However, especially in terms of this kind of planar object, various viewpoint images cannot be obtained. Compared Fig. 12 with Fig. 10, our proposed method can generate better 3D model.

Fig. 13 shows the reconstructed model of KinectFusion[16]. At this time, the KinectFusion which we used was the open source *Kinfu* code in the Point Cloud Library (PCL) from Willow Garage[17]. Using *Kinfu*, Kinect's RGB image, depth image and camera pose of each frame are obtained. Therefore, with the camera pose which KinectFusion estimated, we also obtain reconstructed object shape

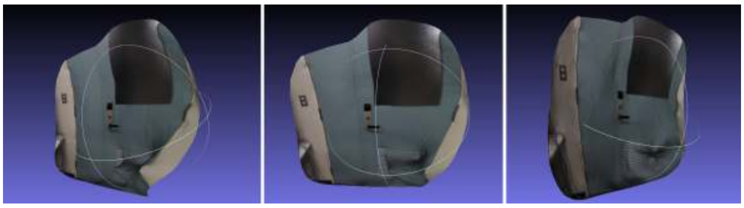


Fig. 12. The 3D model generated by 123D Catch.

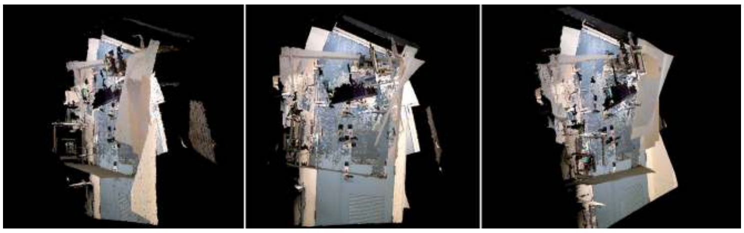


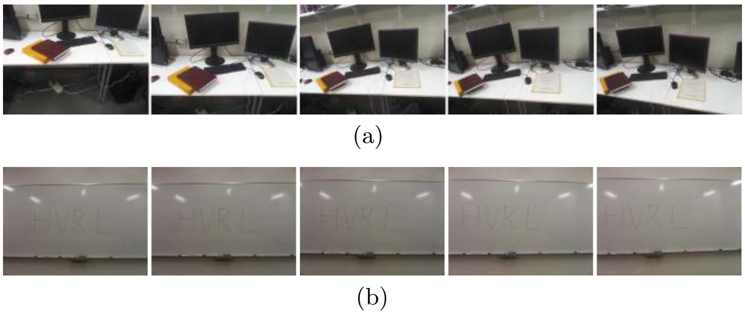
Fig. 13. The 3D model generated by KinectFusion.

represented by colored point clouds. Fig. 13 is virtual viewpoint images of the reconstructed object shape. KinectFusion estimates the camera pose using the alignment of point clouds such as ICP algorithm[18]. However, with this kind of planar object, point clouds alignment is failed, and the estimated camera pose has some error. Therefore, the reconstructed model shown in Fig. 13 is not accurate.

As shown in Fig. 9, Fig. 11, Fig. 12 and Fig. 13, in this situation, our proposed method can reconstruct an accurate 3D model.

3.2 Experiment 2 : On-line camera pose estimation with 3D model

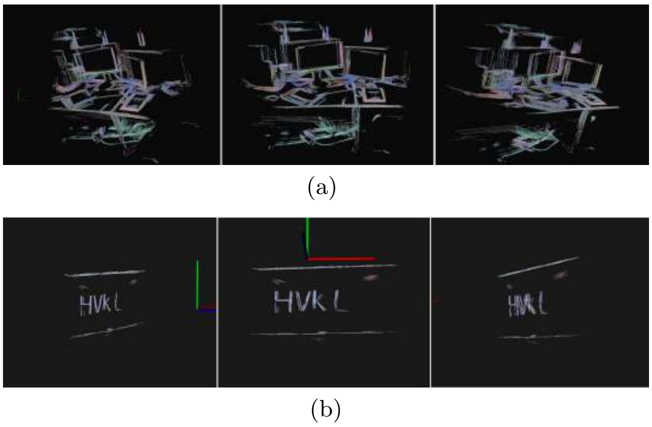
One of the possible applications of the 3D line segment based model generated by using the proposed method is on-line camera pose estimation for mobile AR. We suppose that line segment based model of the target scene for mobile AR is generated by the proposed method. In this model generation phase, the 3D line segments' position in the world coordinate and their Directed LEHF value from their projected 2D line segments on RGB images are stored as 3D line segment database. Then a mobile camera pose of each frame can be estimated on-line process as described below. First, 2D line segments and Directed LEHF values are extracted from the input image. Second, a 3D line segment which has a similar value to the 2D line segment's Directed LEHF value is searched into the 3D line segment database. Then, the 2D line segment from the input image and the 3D line segment in the database are brought to correspondences. These 2D-3D line segment correspondences may have some mismatching. This mismatching is eliminated by RANSAC. Finally, the camera pose of the input image is computed from the 2D-3D matching by RPnL.



(a)

(b)

Fig. 14. Parts of the input RGB images for experiment 2, (a) the desk scene, (b) the white-board scene.



(a)

(b)

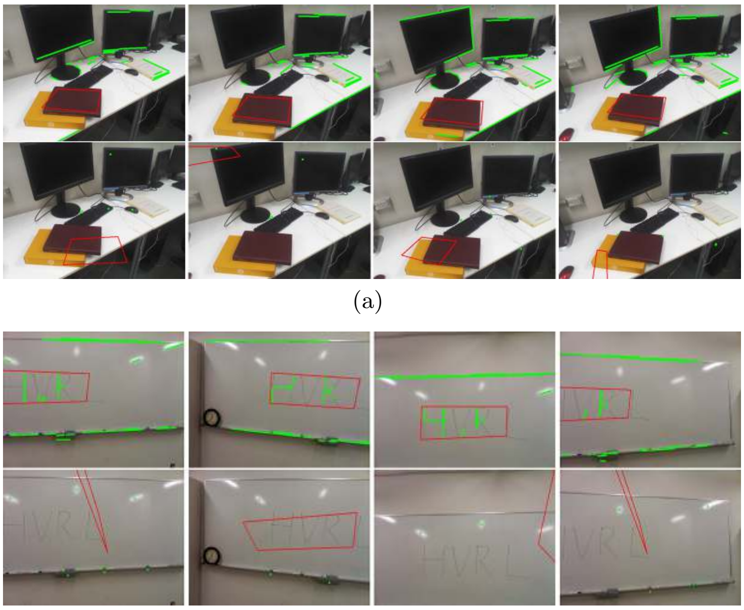
Fig. 15. The 3D line segment based models for experiment 2.

In this experiment, we captured two scenes using Kinect and generated the 3D line segment based model of each scene. One scene is about a desk on which two displays and books are put and another scene is about a white-board on which the word "HVRL" is written. We used 101 frames for the desk scene and 11 frames for the white-board scene. Some of RGB images for the two scenes are shown in Fig. 14. Note that we captured the scenes from the front view of the objects as shown in Fig. 14. Therefore, the 3D line segment based models do not contain line segments from side view of the scenes. Fig. 15 shows generated 3D line segment based models of each scene. With these models, we constructed 3D line segment databases for both scenes. As well as Experiment1, we conducted the same experiment with SIFT feature point matching instead of Directed LEHF matching. Then, we also constructed SIFT feature point databases which contained 3D points' position in the world coordinate and SIFT feature value.

We estimated 27 camera poses of input images for the desk scene and 86 camera poses of input images for the white-board scene with each 3D line segment database and SIFT feature point database. For evaluating the accuracy of

(a) (b)

Fig. 16. The four points of each scene used for measuring re-projection errors.



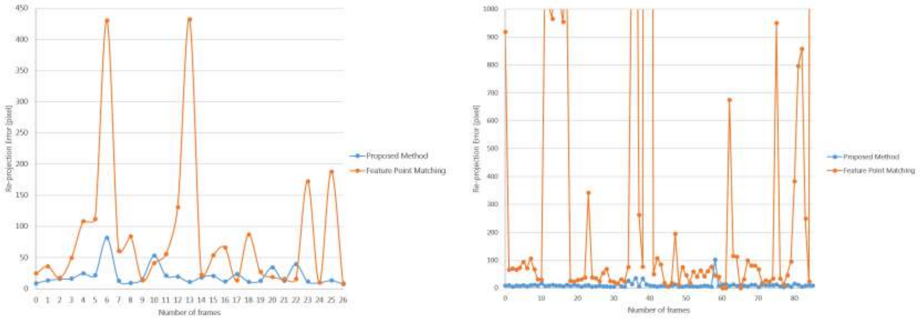
(a)

(b)

Fig. 17. The result images for evaluating the estimated camera pose. (upper row : proposed method, lower row : feature point matching)

the estimated camera poses, we re-projected four points in each scene which are shown in Fig. 16 to the input images, and their re-projection errors are measured. Fig. 17 shows the images which have the re-projected four points. The images on the upper low of Fig. 17(a) and (b) show example images of the result images by our proposed method, and the images on the lower row show example images based on SIFT feature points, respectively. Moreover, the line segments and feature points which used for estimating camera pose are shown on the images in green. These green line segments and feature points include no outlier of RANSAC.

As shown in Fig. 17, in the case of feature point matching, estimated camera poses are not accurate and few feature points are used. On the other hand, our proposed method can estimate the camera poses more accurate and many line segments are used for estimation.



(a) (b)

Fig. 18. Re-projection errors.

Fig. 18 shows the re-projection errors. The camera pose estimation by SIFT feature failed in these situations due to the lack of the correspondences, however, the camera poses which are estimated by proposed method show smaller re-projection errors.

In the case of the desk scene, as shown in Fig. 17(a), the input images are taken from left side view of the desk. However, the 3D line segment based model of the scene contains line segments only from front view of the desk. Therefore, a large viewpoint change is occurred between model and input images. With this, Fig. 17 also shows that feature point matching cannot deal with a change in perspective but line segment matching with our proposed model is robust for this large viewpoint change. This is the advantage of using line segments instead of feature points.

4 Conclusion

We propose a method for generating 3D line segment based model from a RGB-D image sequence captured by RGB-D camera. In this method, 2D line segments are detected by LSD from the RGB image sequence. Each 2D line segments are then associated with depth image for defining a 3D line segment. The line segments are matched between consecutive frames using Directed LEHF for computing camera pose of each frame of the input image sequence. The camera poses of all frames finally generate a 3D model represented by the 3D line segments.

In the experiments, we demonstrate that the proposed method can generate 3D line segment based models even in the case that a few feature points can be detected. We also demonstrate the on-line camera pose estimation for mobile AR application can effectively performed by the use of the 3D line segment based model generated by the proposed method.

References

1. Sturm, P., Triggs, B.: A factorization based algorithm for multi-image projective structure and motion. In: Computer Vision – ECCV'96. Springer (1996) 709–720
2. Lowe, D.G.: Object recognition from local scale-invariant features. In: Proceedings of the Seventh IEEE International Conference on Computer Vision. Volume 2., IEEE (1999) 1150–1157
3. Bay, H., Tuytelaars, T., Van Gool, L.: Surf: Speeded up robust features. In: Computer Vision – ECCV 2006. Springer (2006) 404–417
4. Jain, A., Kurz, C., Thormahlen, T., Seidel, H.P.: Exploiting global connectivity constraints for reconstruction of 3d line segments from images. In: IEEE Conference on Computer Vision and Pattern Recognition, IEEE (2010) 1586–1593
5. Hofer, M., Wendel, A., Bischof, H.: Line-based 3d reconstruction of wiry objects. In: Proceedings of the 18th Computer Vison Winter Workshop. (2013)
6. Hofer, M., Wendel, A., Bischof, H.: Incremental line-based 3d reconstruction using geometric constraints. (2013)
7. Wang, Z., Wu, F., Hu, Z.: Msld: A robust descriptor for line matching. Pattern Recognition **42** (2009) 941–953
8. Hirose, K., Saito, H.: Fast line description for line-based slam. In: Proceedings of the British Machine Vision Conference. (2012) 83.1–83.11
9. Nakayama, Y., Honda, T., Saito, H., Shimizu, M., Yamaguchi, N.: Accurate camera pose estimation for kinectfusion based on line segment matching by lehf. In: Proceedings of the International Conference on Pattern Recognition. (2014) 2149–2154
10. Von Gioi, R.G., Jakubowicz, J., Morel, J.M., Randall, G.: Lsd: A fast line segment detector with a false detection control. IEEE Transactions on Pattern Analysis and Machine Intelligence **32** (2010) 722–732
11. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM **24** (1981) 381–395
12. Fan, B., Wu, F., Hu, Z.: Line matching leveraged by point correspondences. In: IEEE Conference on Computer Vision and Pattern Recognition, IEEE (2010) 390–397
13. Zhang, L., Xu, C., Lee, K.M., Koch, R.: Robust and efficient pose estimation from line correspondences. In: Computer Vision – ACCV 2012. Springer (2013) 217–230
14. Kumar, R., Hanson, A.R.: Robust methods for estimating pose and a sensitivity analysis. CVGIP: Image Understanding **60** (1994) 313–342
15. (Autodesk 123D Catch. http://www.123dapp.com/catch)
16. Newcombe, R.A., Davison, A.J., Izadi, S., Kohli, P., Hilliges, O., Shotton, J., Molyneaux, D., Hodges, S., Kim, D., Fitzgibbon, A.: Kinectfusion: Real-time dense surface mapping and tracking. In: Proceedings of the 10th IEEE International Symposium on Mixed and Augmented Reality, IEEE (2011) 127–136
17. Rusu, R.B., Cousins, S.: 3d is here: Point cloud library (pcl). In: Proceedings of IEEE International Conference on Robotics and Automation, IEEE (2011) 1–4
18. Besl, P.J., McKay, N.D.: Method for registration of 3-d shapes. In: Proceedings of SPIE 1611, Sensor Fusion IV: Control Paradigms and Data Structures, International Society for Optics and Photonics (1992) 586–606