2014 IEEE International Conference on Robotics & Automation (ICRA)
Hong Kong Convention and Exhibition Center
May 31 - June 7, 2014. Hong Kong, China

Line-based 3D Mapping from Edge-Points Using a Stereo Camera

Masahiro Tomono

Abstract— **This paper proposes a method of building a 3D map from straight line segments using a parallel stereo camera. In the proposed method, 3D lines are reconstructed using 3D edge points obtained by edge-based SLAM. This is effective for longitudinal lines, e.g., in a corridor, which are hard to reconstruct due to degeneracy and small disparities. A parallel stereo camera cannot reconstruct horizontal lines in the images, and we reconstruct them using the points of intersection with nearby lines. Furthermore, we propose 1D occupancy cell representation for a 3D line segment to determine its range from multiple view images. Experiments show our approach successfully built line-based maps of indoor environments.**

I. Introduction

A. Motivation

In this paper, we consider a problem of building a map from straight line segments. In many cases, the main structures of man-made environments are comprised of line segments. Line segments have a number of benefits to mapping. They are memory-efficient representation for 3D mapping, and can be components for structured maps. Lines can be the boundaries of planes, and planes can be defined by lines. Furthermore, lines can provide pose constraints to restrict the degrees of freedom. For example, if there is a constraint that two objects are placed on a line, the map will be corrected more accurately and efficiently by pose adjustment.

There have been a number of studies of 3D modeling or mapping using line segments. Most of them have focused on camera motion estimation because lines are detected more stably than corner points in some cases. Line segments can be good components to represent a precise 3D map, but this aspect has yet to be considered well. Motion estimation can be done using only well-detected line segments, and the completeness of line reconstruction is not necessary. However, making a precise map needs complete line sets in the environments. This is not an easy problem and worth considering.

B. Issues

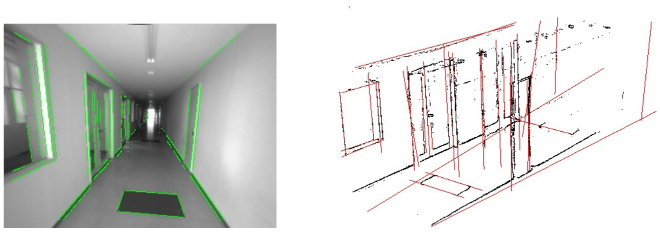
A conventional line reconstruction generates a 3D line from 2D lines detected in two or more images. The methods are basically classified into two kinds: plane-based method and endpoint-based method. The plane-based method uses the plane defined by the camera center and a 2D line on an image. We call this plane *CL-plane* for convenience in this paper. A 3D line is obtained by calculating the intersection of CL-planes over two images. In the case of n images, the intersection can be calculated using least square methods, e.g., applying the singular value decomposition (SVD) to a $n \times 4$ matrix of plane parameters [9], or applying an extended Kalman filter to each image [1], [21]. The endpoint-based method represents a line segment using two endpoints [10], [11]. The endpoints are tracked between images using the epipolar geometry and are reconstructed by triangulation.

These approaches are mathematically clear and useful for both parallel stereo vision and motion stereo. However, the complete reconstruction of lines in the environment is not easy because of the following reasons. One is due to the geometric property of lines. A line cannot be reconstructed when it is on the epipolar line. This is a degenerate configuration in line reconstruction [9]. In the case of a parallel stereo camera, horizontal lines in the image cannot be reconstructed. In the case of motion stereo, the lines parallel to the camera motion cannot be reconstructed. Unfortunately, there are plenty of horizontal lines in man-made environments, and a human or robot encounters such lines when moving on a flat floor or when having a stereo camera parallel to the floor.

Fig. 1 shows an example of line reconstruction using a stereo camera, in which the plane-based method was used. While the vertical lines are successfully reconstructed, the horizontal lines, especially the boundaries between the walls and the floor, are hard to reconstruct. This is because the motion stereo cannot be used for the longitudinal lines. In this case, the angles between any CL-planes are zero or very small and the reconstruction is sensitive to noise. Also, the baseline of the stereo camera is not wide enough to reconstruct the longitudinal lines. A longitudinal line nearly parallel to the camera optical axis is difficult to reconstruct since the endpoint approaches to the vanishing point in the image, around which the stereo disparity is so small as to be sensitive to noise.

Another difficulty is how to determine the range of a line segment. Conventional methods update the end points of a line segment over multiple images by, e.g., projecting the 3D line onto the images and extending the end points to fit it to a 2D line segment on each frame. However, the end point management is quite complicated when there are false positives and negatives in line detection due to bad illumination and

M. Tomono is with Future Robotics Technology Center, Chiba Institute of Technology, Narashino, Chiba 275-0016, Japan. tomono@furo.org



Detected edges

Reconstructed lines (red) and reference map (black)

Fig. 1. An example of line reconstruction in a corridor. The longitudinal lines are hard to reconstruct.

occlusions. In Fig. 1, the end points of the lines were determined by simply merging the line segment in each frame. As a result, some of the lines were extended too long due to the reflection on the floor.

C. Our approach

To cope with the above-mentioned issues, we propose two schemes. First, we employ 3D edge points obtained by edge-based SLAM (EdgeSLAM) developed in our previous work [20]. EdgeSLAM estimates the camera motion and 3D points from image edge points detected in stereo images. In our approach, 3D lines are generated from the 3D edge points by EdgeSLAM instead of being reconstructed from 2D lines. This approach is effective especially for longitudinal lines shown in Fig. 1 since it uses only well-reconstructed edge points, not distant edge points. To cope with horizontal lines in images, which are not reconstructed by a parallel stereo camera, we use an endpoint-based method. It reconstructs a horizontal line using intersection points between the horizontal line and nearby lines.

Second, we propose 1D occupancy cells on a 3D line segment to represent its range. We refer to 1D cell as *line element* or *lixel*. A 3D line consists of a sequence of lixels, and each lixel has an occupancy score which indicates pixels on 2D lines in multiple images are on the 3D line. This scheme represent the range of the 3D line segment in simpler manner than the end points. Also, the occupancy score is based on statistics similarly to the well-known occupancy grid maps, and lixels can represent the range probabilistically.

The contributions of the paper are twofold. One is that it proposes schemes to cope with the problems in line reconstruction mentioned above. The other is that we integrate the schemes to build precise indoor maps using line segments.

II. Related Work

Line-based mapping have been studied in more than two decades [1], [4], [12], [14], [21], [22]. The basic framework is similar; detecting lines from images, tracking lines between images, and estimating the camera motion and 3D lines. In general, line tracking is not easier than corner-point tracking although methods and descriptors have been proposed [2], [16]. Most of the studies on line-based mapping have been focusing mainly on camera motion estimation. On the other hand, our purpose here is to build a structured map using lines. For this purpose, the completeness of line reconstruction is important. We must reconstruct as many lines as possible to build a detailed map.

As mentioned in the previous section, endpoint-based methods for line segment reconstruction have been studied [10], [11]. Given the camera motion, the methods find endpoint correspondences based on epipolar geometry over multiple frames, and calculate 3D endpoints using triangulation and reprojection error minimization. 3D lines are refined by grouping similar 3D lines. Another method uses intersections of 2D lines to find line correspondences and also to estimate the camera motion [13]. These methods are useful when the endpoints are successfully found. However, in the images of a corridor environment such as Fig. 1, for example, longitudinal lines between a wall and a floor (or ceiling) can have no endpoints in the images. This frequently occurs in the context of mobile robot navigation in indoor environments. This makes it hard to use endpoint-based methods for mobile robot applications.

Many studies use geometric constraints to reconstruct lines since line reconstruction is a hard problem [6], [7]. Manhattan world assumption (MWA) is a well-known constraint for building a 3D model of a man-made environment. MWA assumes that an environment consists of horizontal planes (floor and ceiling) and vertical planes (walls). Many man-made environments meet this assumption, and MWA is useful to alleviate errors and ambiguities in line reconstruction. However, non-horizontal or non-vertical lines, which often exist in man-made environments, cannot be reconstructed by this scheme.

Vanishing point (VP) constraints are widely used to estimate line directions in 3D space [15], [17], [18]. VP-based methods are applicable to the lines which are parallel in 3D space. However, it is hard to determine which lines are parallel in 3D space from 2D images. Even if 2D lines are assigned to a VP, some of them can be non-parallel in 3D space. A scheme of strong consistency check is necessary.

A flat floor can be a good constraint. If the height and tilt angle of the camera are known, lines on the floor are easily reconstructed [22]. This is useful especially for wheeled robots moving on a flat floor. However, this constraint cannot be used directly for non-flat environments such as stairs and also for cameras mounted on humans or legged robots.

These geometric constraints are useful to improve accuracy and efficiency, but they need a priori assumptions. Our method needs no assumptions and can re-

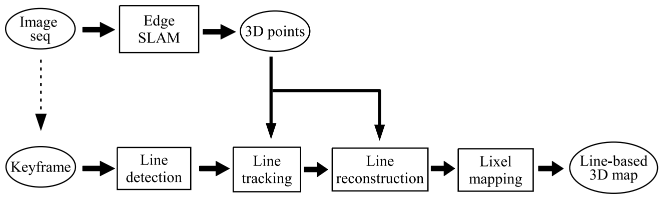


Fig. 2. Flow diagram of the proposed method

construct any lines as long as they are not under degeneracy. It would be better that the robot reconstructs as many lines as possible without any assumptions, and then applies the geometric constraints to complement and refine the map while checking the assumptions using the already-reconstructed lines.

III. Line Reconstruction from Edge Points

A. Basic Idea

Line reconstruction uses EdgeSLAM proposed by our previous work [20]. EdgeSLAM generates 3D edge points by reconstructing image edge points detected by Canny detector [3]. The edge points in each frame are reconstructed based on the parallel stereo vision. The camera motion is estimated by matching the 3D points reconstructed from image I_{t-1} with the 2D points detected in I_t. Based on the obtained camera poses, a 3D map is built by transforming the stereo 3D points from the camera coordinate systems to the world coordinate system.

The method proposed in this paper reconstructs line segments using the camera motion and 3D edge points estimated by EdgeSLAM. While the camera motion and 3D points are estimated for all the frames, line segments are reconstructed from key frames. It is not easy to extract 3D line segments directly from the 3D point cloud due to computational complexity and many outliers. Therefore, we first extract 2D line segments from key frames, and then match the 2D line segments with the 3D edge points to generate 3D line segments. Fig. 2 shows the flow diagram.

2D line segments are detected using LSD developed by Gioi et al. [8]. LSD generates high-quality line segments, but the segments tend to be short since they are divided at every intersection point. Thus, we merge line segments when their endpoints are within 2 pixels and the difference of their directions is within 2 degrees.

EdgeSLAM cannot reconstruct edge points on horizontal lines in the images because it uses the parallel stereo scheme. Thus, horizontal line segments are reconstructed using intersection points as mentioned in Section III-C.

This approach is effective especially for longitudinal lines as mentioned in Section I-C. It can be interpreted that our method selects the part of a line having sufficient disparities by using the edge points well-reconstructed by EdgeSLAM. Also, this approach does not need the endpoints of line segments, which are often missing in the images.

B. Line Reconstruction

The proposed method generates 3D line segments from 2D line segments detected by LSD and 3D edge points generated by EdgeSLAM according to the following procedure.

(1) Line tracking

Line segments are tracked between images using two geometric cues: 3D edge points and epipolar geometry. In the former, we reproject the 3D edge points obtained by EdgeSLAM onto successive two key frames and find the pair of 2D lines which share the maximum number of reprojected points. This process is repeated over all the key frames. The false matches are removed using the normalized correlation of the local patches around the points.

The epipolar geometry is used for the horizontal line segments which have no 3D edge points obtained by EdgeSLAM. The tracking scheme is basically the same with the method proposed in [16]. Each point sampled on a 2D line in a key frame is associated with points on nearby 2D lines in another key frame at the intersections of the epipolar line and the 2D lines. False matches are removed using the normalized correlation of the local patches. We find the pair of 2D lines which share the maximum number of matched points. This method is valid when the lines are not parallel to the camera motion.

We integrate these two schemes as follows. First, 3D point-based tracking is done since this is more efficient and accurate. Then, epipolar-based tracking is done for the line segments which cannot be tracked by the 3D point-based tracking.

(2) Line direction

The direction vector of a 3D line segment is calculated using the 3D edge points. First, we employ RANSAC [5] to find the best direction with outlier removal. It calculates a line direction using two points sampled from the 3D edge points associated with the tracked 2D lines, and finds the maximum consensus among the 3D edge points. We denote the 3D edge points in the maximum consensus by E_d. Then, principal component analysis (PCA) is applied to E_d, and the direction vector $\mathbf{d} = (dx, dy, dz)$ is obtained from the eigenvector having the maximum eigenvalue.

(3) Line representation

In our scheme, a 3D line l is parameterized as $(\mathbf{d}, \mathbf{c}, t, L)$. Here, $\mathbf{d}$ is the direction vector obtained above. $\mathbf{c} = (cx, cy, cz)$ is the origin of the 1D coordinate frame on the line. Parameter t is used to indicate the location on the line, the value of which is 0 at $\mathbf{c}$ and increases

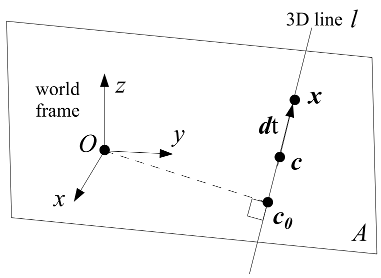


Fig. 3. Line l is expressed by $\mathbf{x} = \mathbf{d}t + \mathbf{c}$. Plane A contains O and $\mathbf{c}_0$, and is perpendicular to l.

along $\mathbf{d}$. L is a lixel representation of l, which is explained in the next section.

Fig.3 illustrates how to determine $\mathbf{c}$. We find the plane A which is perpendicular to $\mathbf{d}$ and which contains the origin of the world coordinate frame. We project E_d onto A and calculate the centroid $\mathbf{c}_0$ of the projected points. Then, we project E_d onto the line $\mathbf{x} = \mathbf{d}s+\mathbf{c}_0$, and calculate the centroid s_g on the line. Now, we obtain $\mathbf{c}$ as $\mathbf{c} = \mathbf{d}s_g + \mathbf{c}_0$.

Based on the parameterization, a point $\mathbf{x}$ on l is represented as

$$\mathbf{x} = \mathbf{d}t + \mathbf{c}. \tag{1}$$

C. Reconstruction of Horizontal Lines

As mentioned above, EdgeSLAM cannot reconstruct edge points on the (nearly) horizontal lines in the images. If a horizontal line has endpoints at the intersections with non-horizontal lines, EdgeSLAM can reconstruct edge points around the endpoints because textures around the intersection points can be rich for EdgeSLAM to find edge point correspondences. Using the reconstructed edge points, a 3D line segment can be reconstructed using the method in the previous section.

However, in some cases, only a small number of edge points are reconstructed around the intersection points, and any line segments cannot be reconstructed. In the cases, we explicitly find the intersection points and reconstruct the line segment between them. Fig. 4 illustrates an example. Here, we assume that non-horizontal line segments are already reconstructed.

(1) Detection of horizontal lines in the images
We collect the 2D line segments which are not reconstructed from 3D edge points yet. In Fig. 4, l_1^2 is such a 2D line segment.

(2) Detection of line intersections
We search 2D line segments in the n-pixel neighborhood of each endpoint of l_1^2 ($n = 5$ in implementation). In Fig. 4, l_2^2 and l_3^2 are such line segments. If the 2D line segments are already reconstructed, they are used to reconstruct l_1^2. The 3D line segments associated with the 2D line segments are stored in L.

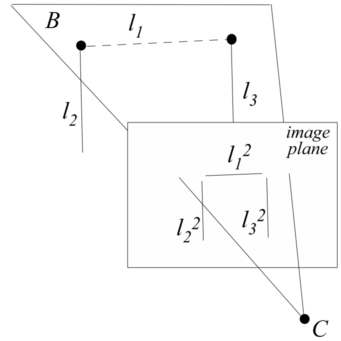


Fig. 4. Horizontal line l_1^2 can be reconstructed from plane B and 3D lines l_2 and l_3. B is a CL-plane generated from C and l_1^2.

(3) Calculation of intersection points in 3D space
A CL-plane B is calculated from l_1^2 and camera center C. Then, we calculate the intersections of B and 3D line segments in L. These 3D intersections corresponds to the intersections of l_1^2 and nearby 2D lines in the images. Note that the 2D intersections might be invisible in the images.

(4) Creation of 3D line segments
A 3D line segment is created by connecting the 3D intersection points. If there are more than two intersection points, multiple line segments are created according to the combination of the points. In this case, the best 3D line segment is selected based on the reprojection error of each 3D line segment to l_1^2.

This method cannot reconstruct a horizontal line if there are no other lines in its neighborhood. This is future work.

IV. Occupancy Cell Representation of a Line

To use a line segment as a component of a map, the range of the line segment must be determined. A line segment is usually represented using its end points. However, a simple rule for updating the end points over multiple frames cause many false positives. To avoid such false positives, a complicated update rule would be needed, which would repeatedly divide and merge line segments. Even if using such a precise update rule, it is hard to address line detection failure caused by occlusions, bad illumination, reflections and shadows.

To cope with this problem, we represent a line segment using a sequence of 1D occupancy cells, or lixels as mentioned in Section I-C. Lixels L for a line l is defined as $\{x_i\}$. A lixel is a short line segment on l. In current implementation, the size of lixel is 2 cm. Index i is a discretized value of t, and the location of x_i in the world frame is determined by Eq. (1). A lixel x_i is a tuple (f_i, o_i), where f_i is the number of fitted points and o_i is the number of observations. f_i and o_i are counted as follows (See Fig. 5).

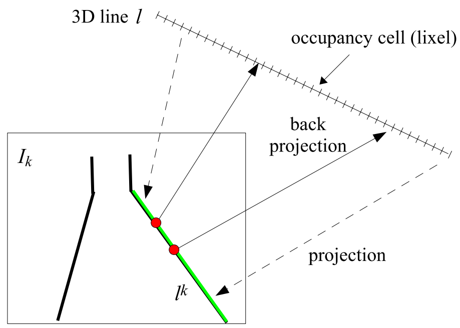


Fig. 5. 1D occupancy cells (lixels) on a 3D line.

We project l onto a key frame I_k in the image sequence and let l^k denote the projected line. The endpoints of l^k are located on the boundary of the image or the vanishing point of l if any. Then, we examine whether or not a 2D line segment exists in the n-pixel neighborhood around each discretized point p_j on l^k ($n = 1$ in implementation). If such a 2D line segment exists and its normal vector is consistent with that of l^k, then p_j is back-projected on to l and the corresponding f_i and o_i are incremented. Otherwise, only o_i is incremented. This process is done for every key frame to accumulate f_i and o_i.

The occupancy score P_i is calculated simply as follows.

$$P_i = \begin{cases} f_i/o_i & \text{if } o_i >= th_1 \\ 0 & \textit{otherwise} \end{cases} \quad (2)$$

The log odds [19] can also be used as the occupancy score.

If the occupancy score exceeds a threshold th_2, the lixel is regarded as a real entity in the world. Thresholds th_1 is 10 and th_2 is 0.4 in implementation. To avoid discarding too many line segments due to bad illuminations, noises, and failures in line tracking, th_2 is set to relatively small value. The range of the line segment is defined by the maximum and minimum value of t at which the occupancy score exceeds th_2.

The advantages of lixel representation are as follows. First, the update of the lixels is easier than that of end points. The range of a line segment is determined by the occupancy score per lixel, and even a complicated line such as dashed line can be represented easily. Second, since the occupancy score is calculated statistically from multiple observations, it is expected to be robust to noise.

V. Experiments

We conducted experiments using Point Grey Research's stereo camera Bumblebee2. Images were captured manually by a walking human. The image size was reduced to 320×240 pixels. The system is implemented in C++ and runs on a laptop PC with Core i7-2960XM 2.7GHz. The stereo camera has the angle of view of 97 [deg].

A. Reconstruction Accuracy

We conducted an experiment on reconstruction accuracy in corridors. As mentioned in Section I-B, corridors have longitudinal lines which are hard to reconstruct due to degeneracy. We compared the proposed method with the plane-based method. The plane-based method was implemented as follows. 2D line segments were detected and tracked by the same schemes with the proposed method. A set of CL-planes are created from the key frames which include a tracked line segment. A best subset of the CL-planes is selected by RANSAC, which calculates a 3D line using two CL-planes sampled randomly and finds the maximum consensus among the CL-planes. Then, SVD is applied to the best subset of CL-planes to find the most feasible 3D line segment. We did not evaluate the endpoint-based method because the endpoints of the longitudinal lines are hard to determine.

Since the ground truth of line maps is hard to obtain, we evaluated line direction errors by comparison with the directions estimated using vanishing points. The vanishing points were detected using Tardif's method [18]. The direction error is calculated as $cos^{-1}(\mathbf{d} \cdot \mathbf{d}_v)$, where $\mathbf{d}$ is the direction vector of a 3D line segment obtained by the proposed method and $\mathbf{d}_v$ is the vanishing direction obtained from the 2D line segments associated with the 3D line segment. We calculated the averages and standard deviations of the direction errors for two types of lines, i.e., 47 longitudinal lines and 92 vertical lines, which were detected in five corridor environments. The stereo camera was moved along the corridors by a walking human.

Table I shows the result. There is no large difference in accuracy between the two methods with respect to the vertical lines. On the other hand, the proposed method has much higher accuracy for the longitudinal lines. Fig. 6 shows images with vanishing directions and reconstructed lines in two environments of the five. As can be seen, the proposed method reconstructed well both the vertical and longitudinal lines while the plane-based method reconstructed well only the vertical lines.

B. Occupancy Scores of Lixels

We conducted experiments to examine how the occupancy scores of lixels works.

Fig. 7 (a) shows line reconstruction of a staircase. This is a challenging environment because the lines of the banister are intersected with those of the background. Many false 3D line segments were generated around the banister. As can be seen in the figure, false 3D line segments decrease as the threshold th_2 increases. An interesting point here is that the line segments of the banister were reconstructed as continuous lines while

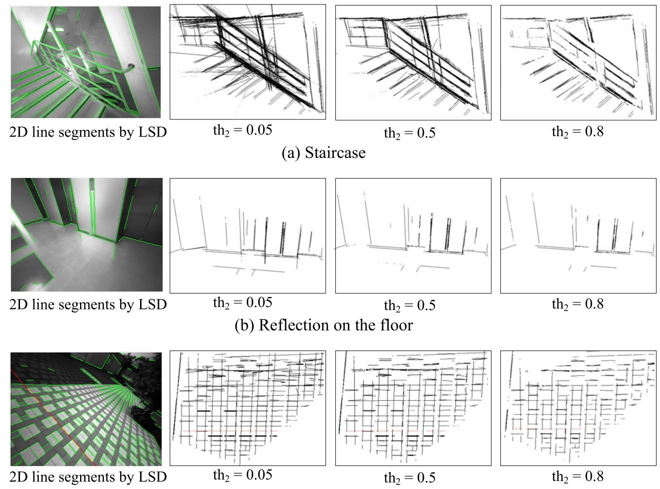
TABLE I

Direction errors evaluated with VPs. The average and standard deviation in [deg] and the number of lines.

	Vertical			Longitudinal		
	avg.	s.d.	#	avg.	s.d.	#
Proposed	2.47	2.21	92	1.58	1.38	47
Plane-based	4.81	8.21	92	12.84	18.77	47

(a) 2D lines assigned to vanishing directions (b) 3D lines generated by the proposed method (c) 3D lines generated by the plane-based method

Fig. 6. For accuracy evaluation in Table I, the directions of the reconstructed lines were compared with vanishing directions.



(a) Staircase

(b) Reflection on the floor

(c) Tiles

Fig. 7. The reconstructed lines vary with threshold th_2 in Section IV. (a) False lines decrease as th_2 increases. (b) False lines by reflection on the floor are removed as th_2 increases. (c) Lixels can represent collinear short line segments as a single line. The red line shows an example.

the 2D lines are divided into pieces in the images. This is because the lixels on the line were sufficiently back-projected from multiple frames.

Fig. 7 (b) shows artifacts due to reflection on the floor. The false 3D line segments due to the reflection diminishes as th_2 increases. Although some of the true line segments also diminishes at $th_2 = 0.8$, many line segments remain visible.

Fig. 7 (c) shows line reconstruction of a tile pattern on a pavement. In this case, short line segments are aligned on a long line. The lixels can represent such a complicated line easily. The red line on the map (and the image) is an example of a 3D line segment represented by lixels. Note that 2D line segments were detected well from the boundaries between the tiles with different colors, but poorly from the boundaries between the tiles with the same color. For this reason, short line segments are aligned on a line at irregular intervals.

C. Line-based Mapping

We conducted experiments of line-based mapping for indoor environments; staircase, corridor, and elevator hall.

(1) Staircase

Edge points were reconstructed by EdgeSLAM from 409 stereo frames, and lines were reconstructed by the proposed method from 45 key frames. Fig. 8 shows the result. The 3D edge points generated by EdgeSLAM are noisy especially around the banisters with cluttered background and also around the steps with near-horizontal lines. The proposed method generated line segments well from such noisy 3D edge points.

(2) Corridor

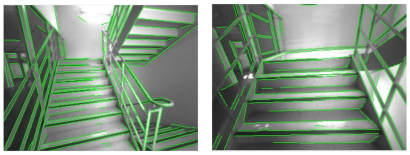
Edge points were reconstructed from 469 stereo frames, and lines were reconstructed from 121 key frames. Fig. 9 shows the result. While the horizontal edges were not reconstructed by EdgeSLAM (rectangles on the floor), the proposed method reconstructed them by the scheme mentioned in Section III-C. The 2D line segments on the boundaries between the ceiling and the walls are hard to detect due to weak contrast and the corresponding 3D line segments are partially missing due to small lixel occupancy scores.

(3) Elevator hall

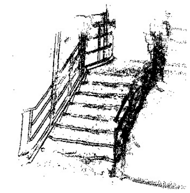
Edge points were reconstructed from 258 stereo frames, and lines were reconstructed from 46 key frames. Fig. 10 shows the result. While most of the horizontal edges were not reconstructed by EdgeSLAM. the proposed method reconstructed them by the scheme mentioned in Section III-C. The horizontal line segment on the ceiling between the pillars was reconstructed with large error, and it is partially visible due to small lixel occupancy scores.

VI. Conclusions

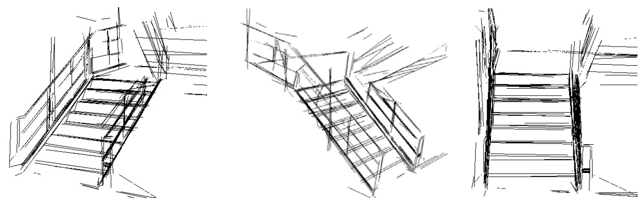
This paper has proposed a method of building a 3D map from straight line segments using a parallel stereo camera. Line segments are important components to represent a 3D map in a structured manner. In the proposed method, 3D lines are reconstructed using 3D edge points obtained by EdgeSLAM. This is effective for longitudinal lines, e.g., in a corridor,



(a) 2D line segments by LSD



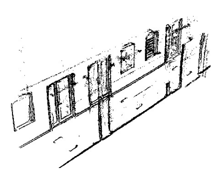
(b) 3D edge points by EdgeSLAM



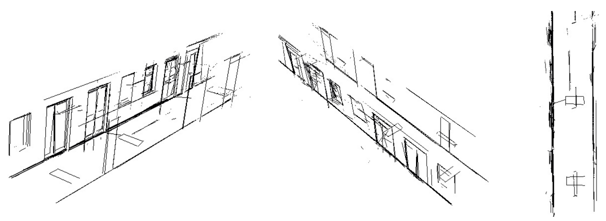
(c) 3D lines by the proposed method from several views

Fig. 8. The result of line reconstruction in a staircase.

(a) 2D line segments by LSD



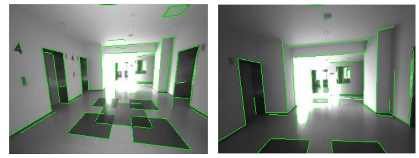
(b) 3D edge points by EdgeSLAM



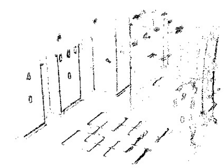
(c) 3D lines by the proposed method from several views

Fig. 9. The result of line reconstruction in a corridor.

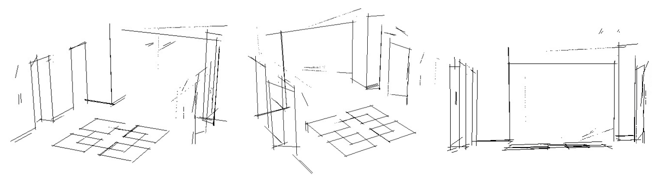
which are hard to reconstruct due to degeneracy and small disparities. Lines cannot be reconstructed by a parallel stereo camera when they are horizontal in the images, and we reconstruct them using the points of intersection with nearby lines. Furthermore, we have introduced 1D occupancy cell representation for a 3D line segment to determine its range from multiple view images. Experiments show our approach successfully built line-based maps of indoor environments.



(a) 2D line segments by LSD



(b) 3D edge points by EdgeSLAM



(c) 3D lines by the proposed method from several views

Fig. 10. The result of line reconstruction in an elevator hall.

References

[1] N. Ayache and O. D. Faugeras: Maintaining Representations of the Environment of a Mobile Robot, *IEEE Trans. on Robotics and Automation*, Vol. 5, No. 6, pp. 804–819, 1989.

[2] H. Bay, V. Ferraris and L. Van Gool: Wide-Baseline Stereo Matching with Line Segments, *Proc. of CVPR2005*, pp.329-336, 2005.

[3] J. Canny: A Computational Approach to Edge Detection, *IEEE Trans. on PAMI*, Vol. 8, No. 6, pp. 679–698 (1986).

[4] M. Chandraker, J. Lim, and D. Kriegman: Moving in Stereo: Efficient Structure and Motion using Lines, *Proc. of ICCV2009*, pp.1741–1748, 2009.

[5] M. Fischler and R. Bolles: Random Sample Consensus: a Paradigm for Model Fitting with Application to Image Analysis and Automated Cartography, *Communications ACM*, 24:381-395, 1981.

[6] A. Flint, C. Mei, I. Reid, and D. Murray, Growing semantically meaningful models for visual SLAM, *Proc. of CVPR2010*, 2010.

[7] Y. Furukawa, B. Curless, S. Seitz, and R. Szeliski: Manhattan-world stereo, *Proc. of CVPR2009*, 2009.

[8] R. G. Gioi, J. Jakubowicz, J.-M. Morel and G. Randall: LSD: A Fast Line Segment Detector with a False Detection Control, *IEEE Trans. on PAMI*, Vol. 32, No. 4, pp. 722-732, 2010.

[9] R. Hartley and A. Zisserman: "Multiple View Geometry in Computer Vision," Cambridge University Press, 2004.

[10] M. Hofer, A. Wendel, and H. Bischof: Incremental Line-based 3D Reconstruction using Geometric Constraints, *Proc. of BMVC2013*, 2013.

[11] A. Jain, C. Kurz, T. Thormaehlen, and H. Seidel: Exploiting Global Connectivity Constraints for Reconstruction of 3D Line Segments from Images, *Proc. of CVPR2010*, 2010.

[12] Y. Kawanishi, A. Yamashita, T.Kaneko and H. Asama: Parallel Line-based Structure from Motion by Using Omnidirectional Camera in Texture-less Scene, *Advanced Robotics*, Vol.27, No.1, pp.19-32, January 2013.

[13] H. Kim and S. Lee: A Novel Line Matching Method Based on Intersection Context, *Proc. of ICRA2010*, pp. 1014–1021, 2010.

[14] G. Klein and D. Murray: Improving the Agility of Keyframe-based SLAM, *Proc. of ECCV2008*, pp.802–815, 2008

[15] J. Kosecka and W. Zhang: Video compass, *Proc. of ECCV2002*, 2002.

[16] C. Schmid and A. Zisserman: Automatic Line Matching across Views, *Proc. of CVPR'97*, 1997.

[17] S. N. Sinha, D. Steedly, and R. Szeliski: Piecewise Planar Stereo for Image-based Rendering, *Proc. of ICCV2009*, 2009.

[18] J. P. Tardif: Non-iterative approach for fast and accurate vanishing point detection, *Proc of ICCV2009*, 2009.

[19] S. Thrun, W. Burgard, and D. Fox: Probabilistic Robotics, MIT Press, 2005.

[20] M. Tomono: Robust 3D SLAM with a Stereo Camera Based on an Edge-Point ICP Algorithm, *Proc. of ICRA2009*, pp. 4306–4311, 2009.

[21] Z. Zhang and O. Faugeras: Building a 3D World Model with a Mobile Robot: 3D Line Segment Representation and Integration, *Proc. of ICPR*, pp. 38–42, 1990.

[22] G. Zhang and I. H. Suh: A Vertical and Floor Line-based Monocular SLAM System for Corridor Environments, *International Journal of Control, Automation and Systems (IJCAS)*, vol.10, no. 3, pp.547-557, 2012.