Proceedings of the Fifth IIEEJ International Workshop
on Image Electronics and Visual Computing 2017
Da Nang, Vietnam, February 28- March 3, 2017

Pedestrian navigation system by aligning 3D reconstructions by Visual SLAM with maps

Daichi Arai† Hiroshi Nagahashi†
†Tokyo Institute of Technology

ABSTRACT

In this research, we propose a navigation system which can estimate a position on a map by aligning a 3D reconstruction result by Visual SLAM on the map. First, the system project the 3D reconstruction result in the gravity direction, and leave only walls of the building using the position of the point cloud and sight line connecting the cameras and the points. Based on the Manhattan-World hypothesis, building walls are transformed into orthogonal line segments using two orthogonal directions in the building. We generate the floor plan of the building by solving the longest path problem of the graph with those line segments as edges. By performing the same processing on the map image and comparing the two floor plans, the system aligns the 3D reconstruction result and the map by proposed geometric hashing method. We show that it is possible to display the current direction to the destination using the 3D reconstruction result of the indoor scene actually obtained by Visual SLAM.

1. INTRODUCTION

In recent years, a navigation system that provides route searching and area information has been an active research topic with widespread use of smartphones. In particular, there has been a growing demand for WYSIWYAS (What You See Is What You Are Suggested) navigation system [1, 2] providing intuitive guidance using AR (Augmented Reality) as shown in Figure 1.

In the navigation system, there are location-based methods using location information such as GPS [3-6] and a vision-based methods [1, 2] using image recognition. The location-based method is widely available where radio waves from the satellite can be accurately acquired, but it cannot be used in high-rise building areas or underground where radio waves from satellites become unstable. On the other hand, the vision-based method can be used anywhere regardless of the location, but in order to use this method, it is necessary to place the marker in the entire area where the marker is desired to be navigated, which takes time and labor to lay. In this research, we propose a pedestrian navigation system that does not require GPS information and markers by aligning the 3D reconstruction result using SLAM (Simultaneous Localization and Mapping) and map information. Also, by displaying route information on the wall using AR, we demonstrate that this system can intuitively present understandable information easily.

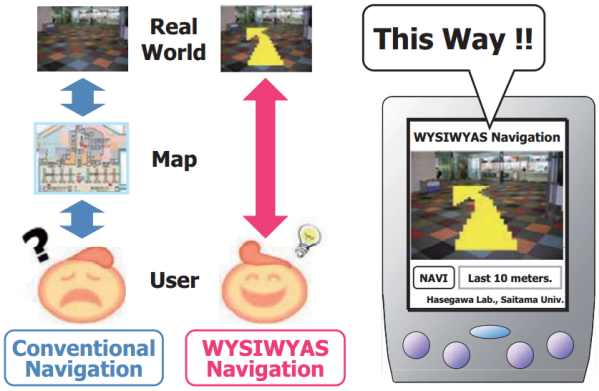


Figure 1.WYSIWYAS concept and example [1, 2]

2. RELATED WORK

Navigation system: As for outdoor location-based methods, some of them are using GPS [3] and others are using UMTS of 3rd generation mobile communication system [4], etc. Both kinds of methods are convenient because they can be used in places where radio waves can be received.

In the case of indoor location-based methods, there are also methods using Bluetooth [5] and methods using Radio Frequency Identification (RFID) [6], but in addition to the mobile terminal for receiving, it is necessary to arrange the transmission equipment.

As a vision-based approach, there is a method using M-CubITS (M-sequence Multimodal Markers for ITS) [1,2]. The method places markers with 0/1 information on the M series, and detects the alignment of the markers from the images taken with the camera. Then it identifies the position and direction of the user by comparing them with the database. The vision-based method has high affinity for WYSIWYAS, and its navigation can be performed only with a mobile phone and it does not require special equipment. However, it is necessary to place markers in advance in the area where navigation is performed.

Visual SLAM: The method of Visual SLAM can be roughly divided into two categories of feature-based method and direct method based on all pixels. PTAM [7] and ORB-SLAM [8] are representative feature-based method. PTAM is a method that enables AR display in real time by performing camera tracking and environmental map creation in parallel processing, but it can be used only in a small environment. Also, since there is only one plane to be detected during initialization,

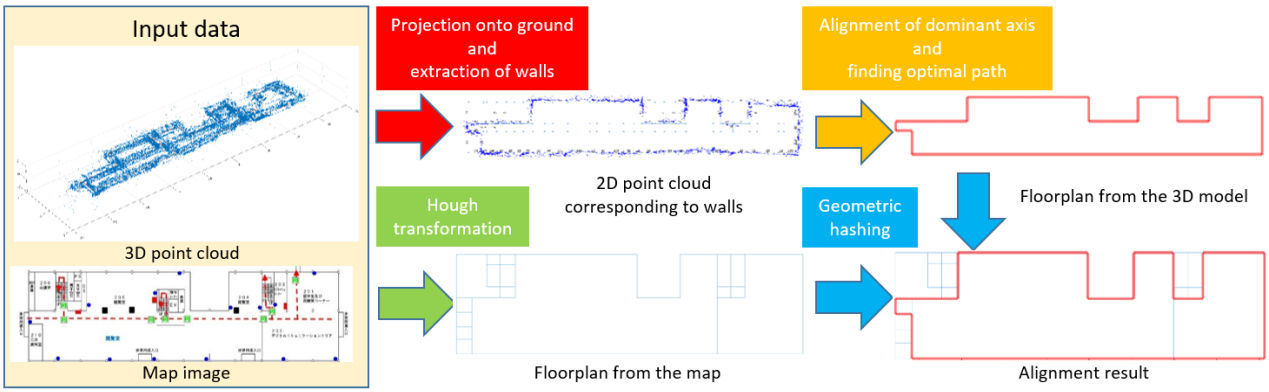


Figure 2. Proposed method overview

PTAM cannot display AR on multiple planes. On the other hand, ORB-SLAM can be used by loop closing for wide-area environments. However, there is no function to display Augmented Reality in ORB-SLAM, and it is assumed that surfaces are well-textured and roughly Lambertian.

LSD-SLAM [9] is a representative direct method. LSD-SLAM can reconstruct dense 3D points by estimating the camera position and orientation so that photo-consistency of all the pixels on the edge is maximized. It can be used in a wide environment due to loop closing, but the accuracy of camera position estimation is generally lower than feature-based method.

The advantage of LSD-SLAM is that a number of points to be reconstructed is large, but it may sometimes fail in tracking due to scenes. So in this research we conducted an experiment using ORB-SLAM.

3D to 2D alignment: The paper most related to our proposed method is a literature of alignment of 3D point clouds to overhead images [10]. It projects the SfM reconstruction onto the ground plane to create a 2D point cloud, and geometrically transforms the 2D point cloud onto the satellite image to align it. Two cost functions are defined, i.e. edge cost and free-space cost. The edge cost represents the distance from the 2D point cloud after geometric transformation to the wall on the overhead image. In contrast, the free-space cost represents the absence of an object on the camera's sight. Those two cost functions enable to estimate the geometric transformation using the coarse-to-fine minimization.

The method [10] addressed a particularly simple case where the 3D reconstruction and the floor plan are perfectly accurate. However, these two patterns do not match in practice because of the error accumulation of Visual SLAM. In addition, we assume that a target map is handmade, so it is impossible to accurately align the target map with the reconstructed model by the method [10].

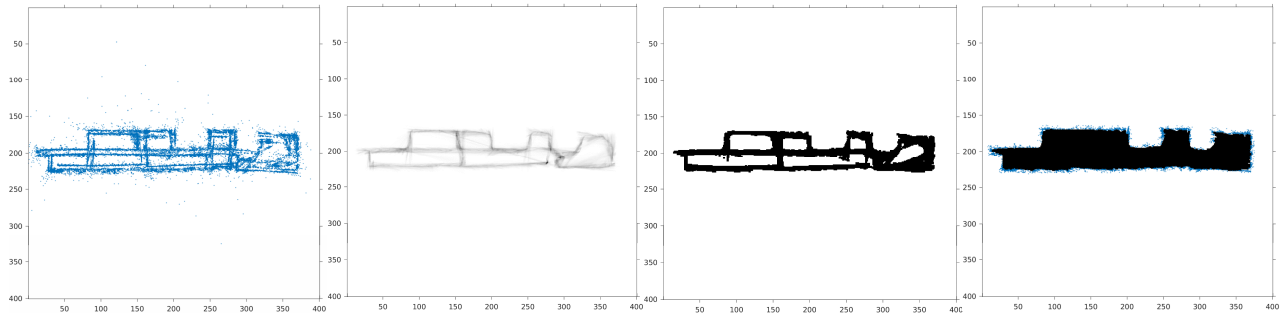
In our research, we aim to align one map to one 3D model. In this context, a method has been also proposed that deals with multiple SfM reconstructions at each place of a map [20]. The method estimates the position of the candidate on the map of the SfM reconstructions using the annotated indoor map and Google image search, and then optimizes the position, direction and scale of the 3D models with respect to the shape of the floor plan of the map. Further, from the time series of the photograph, considering which order the photographer circulated in the room, it estimates better position, direction and scale of the 3D model.

3. SYSTEM OVERVIEW

The system is divided into preparatory stage and execution stage. In the preparatory stage, the entire indoor scene is captured and a 3D restoration is performed by Visual SLAM. After that, the proposed system automatically aligns the 3D model and the map. In the execution stage, the system finds the shortest path from a destination entered on the map and displays route information on walls using AR. Users are able to arrive at their destination if they proceed as suggested.

3.1. Proposed method overview

Figure 2 shows an overview of the proposed method. We use 3D point cloud and map image obtained by Visual SLAM as input. First, in order to compare the 3D point cloud with the 2D image, the 3D point cloud is projected in the gravity direction and converted into a 2D point cloud. Next, by using the fact that the space between the camera and points must be free-space, it is possible to extract points from the 2D point cloud that correspond to building walls. Then, based on the Manhattan-World hypothesis that planes of a building are arranged perpendicularly or parallel to three axes orthogonal to each other, we estimate two orthogonal directions in the building and transform the 2D point cloud into orthogonal line segments. The system constructs a graph with those line segments as edges, and generates a floor plan by finding the optimum route. On the other hand, a floor plan of the map image is generated by using Hough transform. By finding corresponding points using geometric hashing, we align the two floor plans.



(a) 2D point cloud (b) Ray image (c) Truncated ray image (d) core free-space

Figure 3. Example of ray image. If objects such as columns and shelves are present in the indoor scene, the truncated ray image cannot cover the whole indoor scene as shown in Figure 3(c). In that case, the part surrounded by core frees-pace is handled as core free-space as well as shown in Figure 3(d).

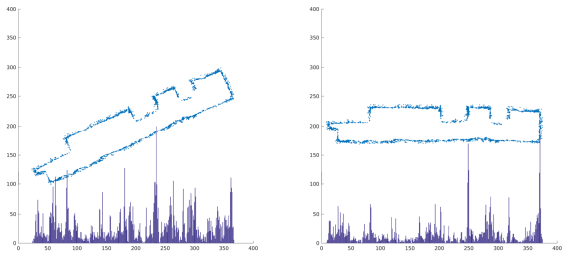


Figure 4. Estimation of wall directions.

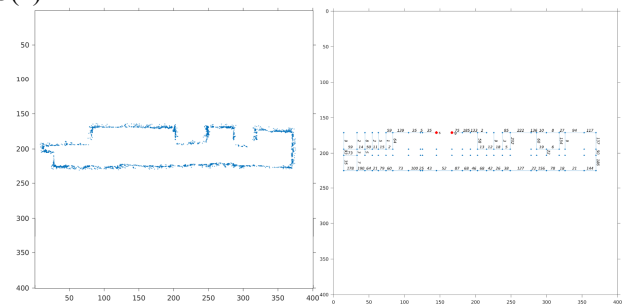


Figure 5. Floor plan reconstruction as longest path

3.2. Generation of 2D point cloud

Devices such as smartphones have a function of recording the gravity direction by an acceleration sensor, but in the case of using a general camera, it is necessary to estimate the gravity direction. Here, we use the assumption that the horizontal direction in the image coordinate system is perpendicular to the gravity direction. The horizontal direction vectors in the image coordinate system are extracted for all images, and the vectors most perpendicular to all the horizontal vectors are used as the gravity direction [11]. Using the estimated gravity vector, each 3D point is projected in the gravity direction to create a 2D point cloud.

3.3. Extraction of walls included in 2D point cloud

Since the 3D reconstruction by Visual SLAM usually includes not only building walls but also points that are not walls. To eliminate not-wall-points, we use the fact that the path between a camera and any point visible to the camera should be free of occluders. We first create an image called ray image. This image is created by tracing every ray from each camera location to all points that are deemed visible to the camera during the Visual SLAM computation. Each pixel in the ray image along a line segment from the camera is incremented. Once all source locations are processed, each pixel in the ray image will have a value of k, the number of rays that pass through that pixel or 0 if no rays pass through it. Next, using the ray image, the system extract walls from the 2D point cloud. An area where the pixel value of the ray image is larger than a certain threshold is defined as a core free-space. Removing points contained in the core free-space leaves only the building walls. In the implementation, the threshold is 0.5σ where σ is the average of the ray image over pixels with non-zero values.

3.4. Estimation of wall directions

Normally, a 3D point cloud reconstructed by Visual SLAM and a map are not perfectly the same shape. Therefore, in order to obtain the approximate shape of 2D point cloud, we utilize Manhattan-World hypothesis [23] that the planes constituting the building are arranged perpendicular or parallel to three orthogonal axes. Based on the Manhattan-World hypothesis, we estimate two directions corresponding to the orientation of the walls, and generate a graph composed of orthogonal line segments from the 2D point cloud. If the method using the 3D point cloud obtained by PMVS [12] or a depth sensor [13], the direction of the normal of each point has already been known, but in this method only the ORB-SLAM with monocular camera is available. Hence, we obtain the orientations of the walls based upon the assumption that a wall projected in the gravity vector becomes a line segment on a plane. While rotating the 2D point group from 0 to 90 degree, we project the 2D point cloud on each of two orthogonal axes and create a histogram showing the number of points for each axis. As shown in Figure. 4, when the sum of the entropies of the two histograms becomes the smallest, the directions of the two axes coincide with those of the walls.

3.5. Floor plan reconstruction as longest path

After estimating the wall direction, we generate a graph composed of orthogonal line segments from the 2D point cloud. Utilizing the fact that when projecting a 2D point cloud onto two wall directions, the local maximum of the histogram can be thought to be a wall. In the implementation, we detected candidate walls by means of 1D Mean Shift [15]. Thereafter, we assigned each point to the nearest line segment, and constructed a graph in which the cost of each edge is the number of points assigned to the line segment as shown in Figure. 5. However, as for edges passing through core free-space and those with cost 0, we deleted them.

The graph also includes walls that do not actually exist. Since the floor plan of an indoor scene is represented by a closed loop, solving the longest path problem of the graph enables to extract the wall contained in the closed loop. In oreder to determine the start point and the end point, exhaustive search method is used as in the method of [16]. Solving the longest path problem by exhaustive search can extract only the correct wall.

3.6. Alignment of floor plan by geometric hashing

First, in order to compare the floor plan generated from the 3D point cloud and the map image, we generate a floor plan from the map image as well. Assuming that the image coordinates and the wall direction correspond with each other, we binarize the input map image and extract only line segments that coincide with the wall direction using mathematical morphology [22] and probabilistic hough transform [17].

For the floor plan generated from the map image, wall candidates are detected by means of 1D Mean Shift as with generation of 2D point cloud in Section 3.5. Each pixel is assigned to the nearest line segment, and a graph is constructed in which the cost is given as the number of points assigned to the line segment. However, edges with cost 0 are deleted.

After that, we align the two floor plans using geometric hashing [19]. In an ordinary geometric hashing, feature points are extracted from the image, and invariants are calculated only using the positions of the feature points. A hash table is created from these invariants, and the input image is recognized by voting. Moreover, in this study, it is possible to utilize not only the positions of the 2D points but also the connection of the points through the walls and the limitation of wall directions based on Manhattan-World hypothesis. Therefore, in order to align the two graphs with different shapes, we extract all the loops included in the floor plan of the map, and calculate the invariants from those loops. Then we create a hash table to align with the floor plan created from the 3D point cloud.

In this process, loop extractions from the floor plan were performed by [18]. By finding the minimum spanning

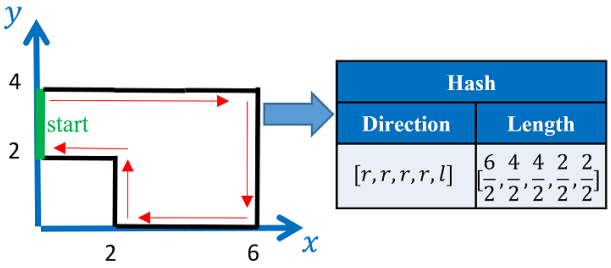


Figure 6. Example of creating a hash table. Find invariants for all loops and all starting edges in the map.

tree from the graph of the map, the maximum number of loops is represented by $2^n - 1$ with respect to the number n of edges deleted at that time. For each loop, we register the invariant directions and distances of all edges around the loop are registered as shown in Figure 6. At this time, the direction is expressed as left and right two directions, and the distance is expressed as a ratio with respect to the starting edge, so that it is invariant to rotation, translation and scale changes. After creating the hash table for the map, we also obtain the invariants for the floor plan of the 3D point cloud. After that, vote for the loop in which the invariants of rotation is perfectly matched and the invariants of the distance is the closest. By finding the loop that got the most votes, we align the 3D point cloud and the map.

3.7. Navigation system

Here we describe a method of finding the shortest route to the destination on the map entered by a user at execution stage and presenting the route information on the wall with AR.

Firstly, the system obtains the position on the 2D point cloud that corresponds to the destination on the map. Based on the assumption that the destination is near the wall, find the nearest edge from the destination in the graph of the map and find the corresponding edge in the graph of the 2D point cloud. After that, by solving the shortest path tree of the graph from the edge corresponding to the destination, the direction to the destination at each edge is obtained.

Next, in order to display AR on the walls, the system obtains the points at the four corners of the wall in the camera coordinate. The system projects the 3D point cloud on gravity vector and obtains the height of the floor and ceiling by 1D Mean Shift. In addition to this, we use the position on the horizontal plane of the walls to find the position of the four corners of all walls in 3D space. On the other hand, since the position and orientation of the current camera can be obtained by localizing ORB-SLAM, it is possible to estimate the positions of the four corners of all walls in the camera coordinate system using projective transformation. Finally, navigation using AR is performed by drawing animations on the screen using alpha blending.

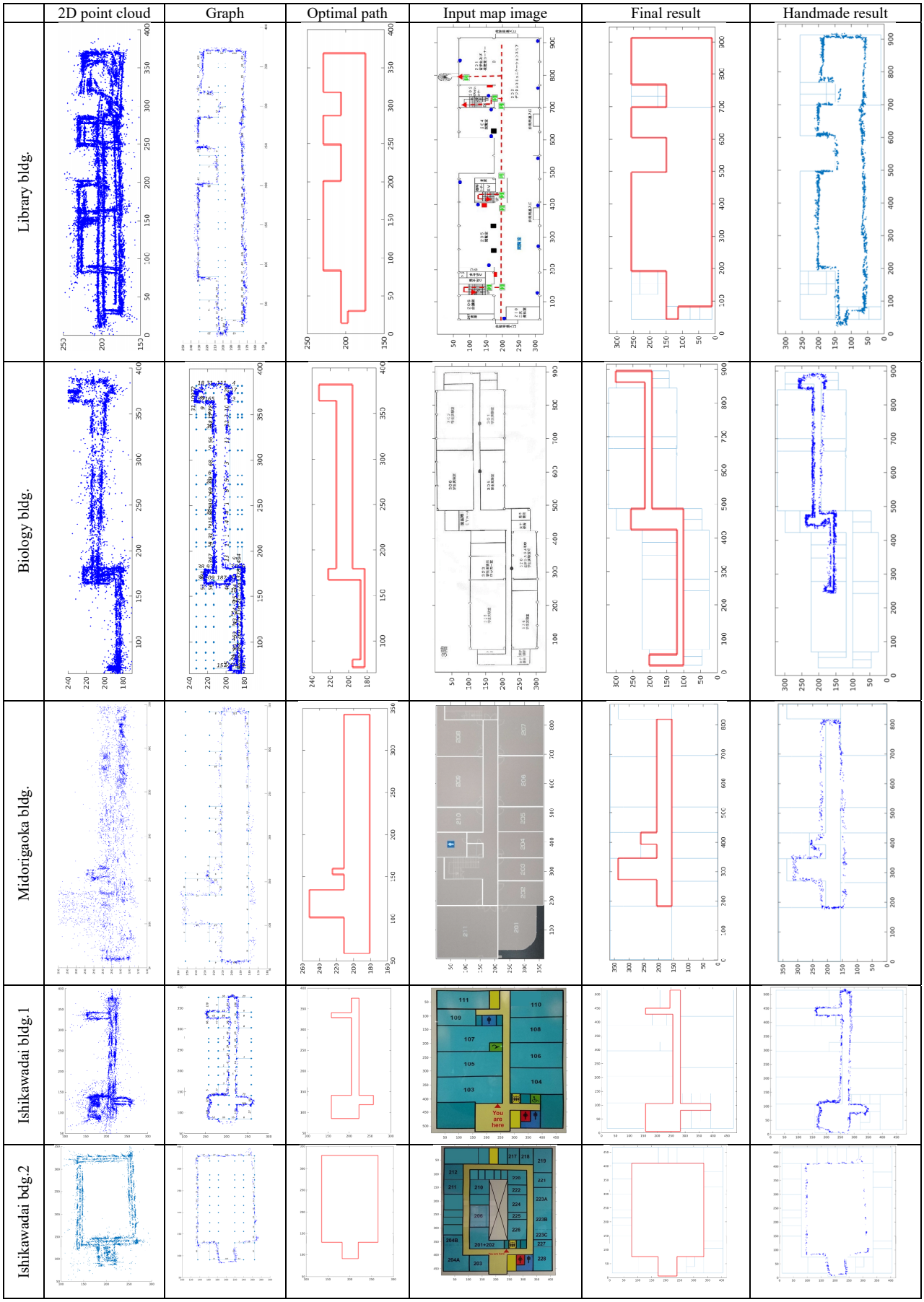


Figure 7. Experimental results.

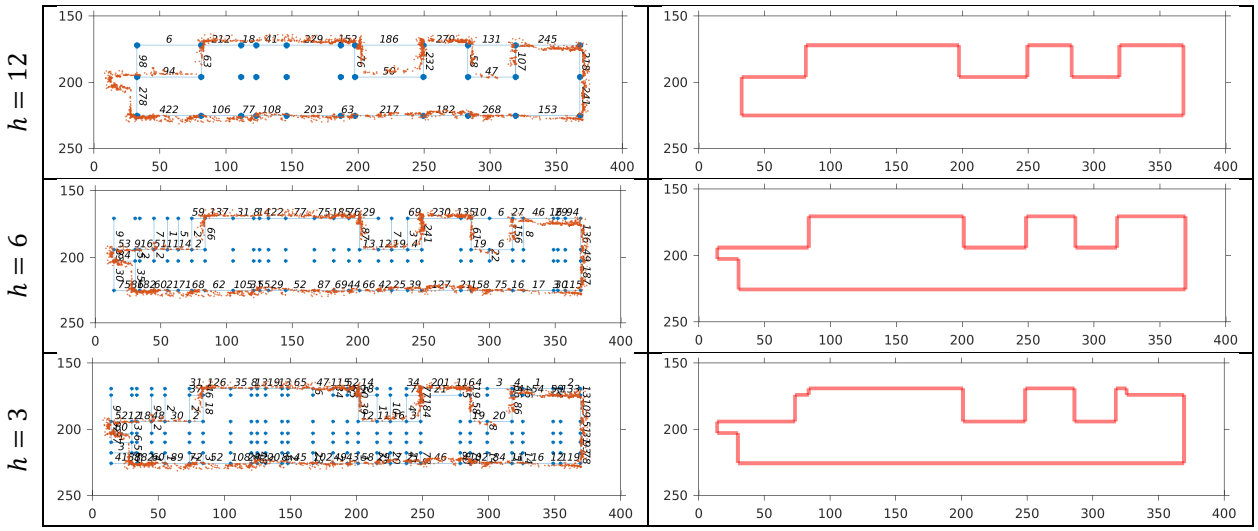


Figure 8. An example of changing the Mean Shift bandwidth.

4. EXPERIMENTAL RESULTS

We have tested our algorithm on five real datasets, as shown in Figure 7. All the experiments have been performed on an Intel Core i5-4210U (1.70GHz) machine with 8.0GB RAM. MATLAB and C++ with ROS [21] are used for the implementation. We utilize ORB-SLAM [11] which is a state-of-art feature-based SLAM.

As shown in the 6th column of Figure 6, the 3D point clouds reconstructed by ORB-SLAM and the maps are not perfectly the same shape. It was because the shape of the map was different from the actual shape in the Library bldg. dataset, and because the surfaces were not well-textured and roughly Lambertian so 3D reconstruction was not performed correctly in the Biology bldg. dataset. However, our method enables to align the two inputs with different shapes.

Table 1 summarizes some characteristics of the datasets in our method. N_p is the number of reconstructed points , and N_c is the number of input images.

h is the Mean Shift bandwidth when generating the floor plan of the 2D point cloud. It represents a ratio to the 2D point cloud normalized to 400 × 400. Figure 8 shows an example of changing the Mean Shift bandwidth. Our method is sensitive to this parameter. If h is large, the details of the shape of the floor plan are ignored, and if h is small, the wrong shape is obtained due to the error of ORB-SLAM. When the parameter is proper, the correct floor plan can be detected by solving the longest path problem. Automatic estimation of the Mean Shift bandwidth is a future work.

N_L is the maximum number of all loops included in the graph generated from the map image. In the implementation, a hash table is created for all loops. Since the total number of loops increases exponentially as the map becomes more complex, time complexity and space complexity also increase. Therefore, depending on the map image, the computational complexity may be enormous.

	N_p	N_c	h	N_L
Library	57045	851	6	65535
Biology	20522	484	3	2097151
Midorigaoka	8337	220	6	32767
Ishikawadai-1	17683	472	6	8388607
Ishikawadai-2	17427	321	10	8388607

Table 1. Characteristics of the datasets

Figure 9 shows an example of a result of extracting line segments from the input map image. From top to bottom, an input map image, a line-segment extraction image and its reformation image are given. The extraction of line segments were performed by binarization, morphological operations and probabilistic hough transformation.

Our method can extract only the walls from the input map image. However, these processes are sensitive to the parameter changes and it is necessary to adjust the parameters depending on the input image. A future work is to automatically extract the floor plan from the map image using such as a neural network.

Finally, the shortest path tree of the graph to the destination is shown in Figure 10, and an example of the user interface of the pedestrian navigation system is shown in Figure 11. The arrows in Figure 10 coincide with the directions of the route information in Figure 11. The user interface displays route information on walls using AR, so the user can arrive at the destination if he proceeds as suggested.

5. CONCLUSION

In this research, we proposed a markerless vision based pedestrian navigation system by aligning the 3D point cloud reconstructed by Visual SLAM with the map image. In addition, we also proposed a geometric hashing method that can align 2D points with map images with different shapes.

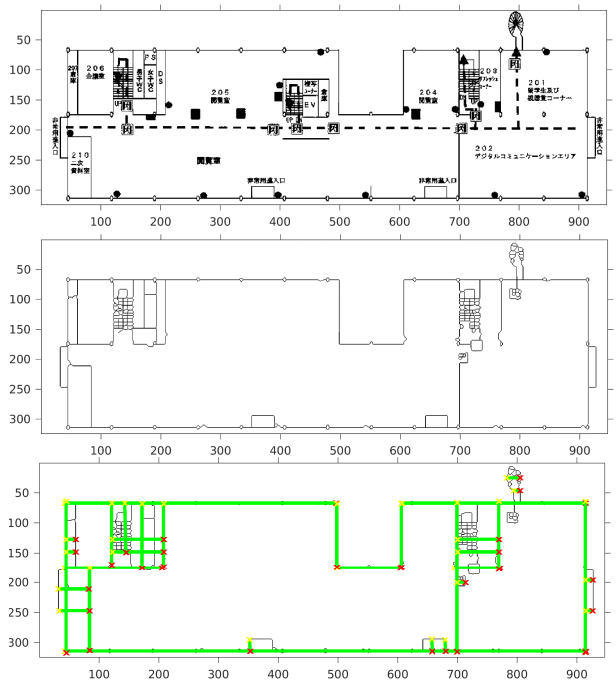


Figure 9. Line segments extraction from the map

We believe that our method is superior to existing methods in the following three points. Firstly, this system enables to perform pedestrian navigation without special equipment in indoor scenes where GPS cannot be used. Secondly, unlike other Visual SLAM, it is possible to detect multiple planes and display AR. Thirdly, even if the 3D model and the shape of the map are different shapes, our method can align them.

However, our system has limitations that the experiment environment has to satisfy the Manhattan-World hypothesis and the user needs to adjust appropriate parameters. Future work should handle indoor scenes composed of three or more wall directions and automatically adjust the parameters. It would also be interesting to implement easy-to-understand navigation system using 3D characters.

6. REFERENCES

[1] T. Manabe, S. Yamashita, T. Hasegawa: “On the M-CubITS Pedestrian WYSIWYAS Navigation System”, IEEE Conference on Intelligent Transportation Systems, pp.793-798 (2006).

[2] T. Manabe, T. Hasegawa, T. Serizawa, N. Machida, Y. Yoshida, and T. Fujiwara: “New Types of Markers and the Integration of M-CubITS Pedestrian WYSIWYAS Navigation Systems for Advanced WYSIWYAS Navigation Environments”, IEICE Transaction on Fundamentals, Vol.E99-A, No.1, pp.282-296 (2016).

[3] M. Arikawa, S. Konomi, K. Ohnishi: “Navitime: Supporting Pedestrian Navigation in the Real World”, IEEE Pervasive Computing archive, Vol.6, Issue 3, pp.21-29 (2007).

[4] M. Umlauft, G.Pospischil, G. Niklfeld, E. Michlmayr:” LOL@, A mobile tourist guide for UMTS”, Journal of Information Technology & Tourism, Vol.5, pp.151-164 (2003).

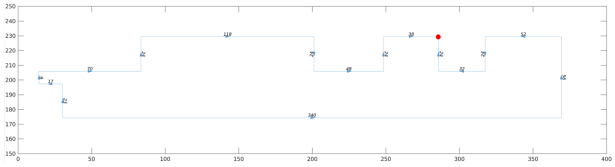


Figure 10. The shortest path tree of the graph to the destination

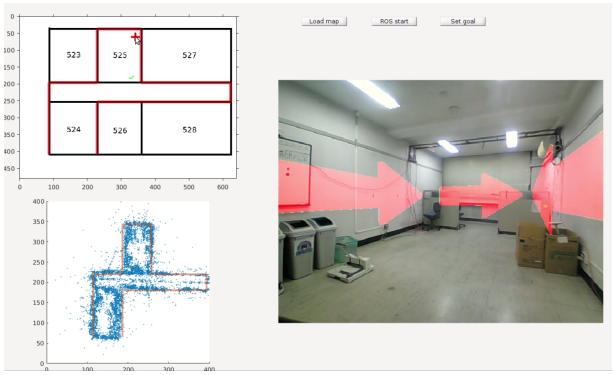


Figure 11. An example of the user interface of the pedestrian navigation system

[5] M. Sakurai, M. Matsubashi:” Pedestrian navigation using INFO SIGN”, NEC technical report, No.1, pp.53-56 (2008).

[6] M. Bessho, S. Koshizuka, K. Sakamura:” A space-identifying ubiquitous infrastructure and its application for tour-guiding service”, Proceedings of the 23th Annual ACM symposium on Applied computing, pp.1616-1621 (2008).

[7] G. Klein, D. Murray:” Parallel Tracking and Mapping for Small AR Workspaces”, Proceedings International Symposium on Mixed and Augmented Reality, pp.225-234 (2007).

[8] R. Artal, J. Tardos: “ORB-SLAM: Tracking and Mapping Recognizable Features”, Robotics: Science and Systems Workshop on Multi View Geometry in Robotics (2015).

[9] J. Engel, T. Schops, D. Cremers, “LSD-SLAM: Large-scale direct monocular SLAM,” European Conference on Computer Vision (2014).

[10] R. Kaminsky, N. Snavely, S. Seitz and R. Szeliski: “Alignment of 3D point clouds to overhead images”, Computer Vision and Pattern Recognition Workshops (2009).

[11] R. Szeliski:” Image alignment and stitching: A tutoria”, Foundations and Trends in Computer Graphics and Computer Vision, 2(1) (2006).

[12] Y. Furukawa and J. Ponce. PMVS. http://www.di.ens.fr/pmvs/ (2010).

[13] S. Ikehata, H. Yan, Y. Furukawa: ”Structured Indoor Modeling”, International Conference on Computer Vision (2015).

[14] M. Pollefeys et al: ”Detailed real-time urban 3d reconstruction from video”, International Journal of Computer Vision, Vol.78, No.2-3, pp.143–167 (2008).

[15] D. Comaniciu, P. Meer: “Mean shift: A robust approach toward feature space analysis”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.24, No.5, pp.603–619 (2002).

[16] R. Cabral, Y. Furukawa: “Piecewise Planar and Compact Floorplan Reconstruction from Images”, Computer Vision and Pattern Recognition (2014).

[17] N. Kiryati, Y. Eldar, A. Bruckstein: “A probabilistic hough transform”, Pattern Recognition, Vol. 24, No. 4, pp. 303-316 (1991).
[18] L. Chai: “Algorithmic approaches to circuit enumeration problems and applications”, Flight Transportation Laboratory Report R82-7 (1982).
[19] Y. Lamdan, H. Wolfson: “Geometric hashing: a general and efficient model based recognition scheme”, International Conference on Computer Vision (1988).
[20] R. Brualla, Y. He, B. Russell, S. Seitz: “The 3D jigsaw puzzle: mapping large indoor spaces”, European Conference on Computer Vision (2014).
[21] Robot Operating System, http://www.ros.org/ (2007).
[22] L. Pitas, A. Venetsanopoulos: “Morphological shape decomposition”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol .12, No.1, pp.38−45 (1990).
[23] Y. Furukawa, B. Curless, S. M. Seitz, R. Szeliski: “Manhattan-world stereo”, Computer Vision and Pattern Recognition (2009).