Using Semantic Maps for Room Recognition to Aid Visually Impaired People

Qiang Liu, Ruihao Li, Huosheng Hu, Dongbing Gu

School of Computer Science and Electronic Engineering, University of Essex, Colchester CO4 3SQ, UK
Email: qliui@essex.ac.uk, rlig@essex.ac.uk, hhu@essex.ac.uk, dgu@essex.ac.uk

***Abstract*—Millions of people in the world suffer from vision impairment or even vision loss. Guide sticks and dogs have been deployed to lead them around various obstacles. However, both of them are not capable of interacting with human users who normally rely on conceptual knowledge or semantic contents of the environment. This paper first builds a 3D semantic indoor environment map with an RGB-D sensor. Then, the map is used for room recognition during the revisits based on appearance by applying a convolutional neural network. Representative objects extracted from the semantic map are used to diagnose and eliminate errors during room recognition. The proposed method result in a 97.8% accuracy even with lighting condition and small object location changes.**

***Keywords*—Semantic map, room recognition, visually impaired people, convolutional neural network, HMI.**

I. Introduction

Nowadays, 285 million people are estimated to be visually impaired worldwide, among which 39 million suffer from total blindness [1]. Guide sticks and dogs have been deployed to lead blind people around various obstacles. However, the guide sticks are too simple to be effectively used and the guide dogs are expensive to be trained. Moreover, both of them are unable to interact with human users with conceptual knowledge or semantic contents of the environment. Thus, it remains a major challenge for blind people to navigate independently, especially in unfamiliar indoor environments.

Traditional maps, namely geometric ones and topological ones, are only navigation oriented, which enable mobile robots to navigate around and plan a path for reaching a goal [2]. However, these maps are inadequate for guiding blind people who normally need semantic information interpreted from scenes. To build a robotic system to help blind people, a semantic map is the preliminary requirement. In a semantic map, nodes representing places and landmarks are named by linguistic words [3]. More specifically, a semantic map can be regarded as a hybrid map, which contains geometric description, topological connectivity and semantic interpretation [4]. It provides a friendly way for service robots to communicate with users.

A straightforward method for room recognition is based on appearance. However, not all the objects remain the same locations, e.g. curtain, chair and pet. Furthermore, the lighting conditions can vary at different times. In order to eliminate errors caused by these constantly changing conditions, representative objects extracted during semantic mapping are used to further verify our room recognition results.

Our ultimate goal is to build a robotic system to help visually impaired people navigate indoor environments and improve the quality of their life. The system consists of a wearable device and a computer which are connected through wireless communication. The device is used for collecting RGB-D images, sending them to the computer and providing voice prompts, whereas the computer is used for data processing, e.g., semantic mapping, relocalization and path planning. This paper presents our initial result on semantic mapping and topological relocalization.

The rest of the paper is organized as follows. Section II reviews related literature works. Our 3D semantic mapping and room recognition methods are detailed in Section III. Subsequently, experiment results are presented and discussed in Section IV. A brief conclusion and future work are given in the last section.

II. Related Works

Recently, semantic mapping has become a popular research topic and drawn much attention in the robotics field. In [5], models of indoor Manhattan scenes were acquired from individual images generated from a 2D camera and then assigned with semantic labels. Tian *et al.* addressed door modeling and detection problem to assist blind people to access unfamiliar indoor environments [6]. Wu *et al.* [7] and Neves dos Santos *et al.* [8] employed 2D visual data to tackle the problem of place recognition in semantic mapping. SLAM and object recognition with monocular cameras were presented by Civera *et al.* [9] and Riazuelo *et al.* [10], respectively.

Due to the availability of low-cost and light-weight 3D point cloud capturing devices, such as stereo cameras and RGB-D sensors, mapping environments into 3D point clouds and extracting semantic information from them have become a new trend [11], [12], [13], [14] Compared to 2D images, 3D point clouds overcome the limitation in the data-stream itself by providing additional depth data. In fact, 3D point cloud is a natural way to recognize objects in a 3D world, especially in terms of a large variety of goods in our daily life [15].

Convolutional Neural Networks (CNN) has been an approach widely used in the computer vision and machine learning domain and outperformed other methods in terms of visual recognition, classification and detection tasks [16].

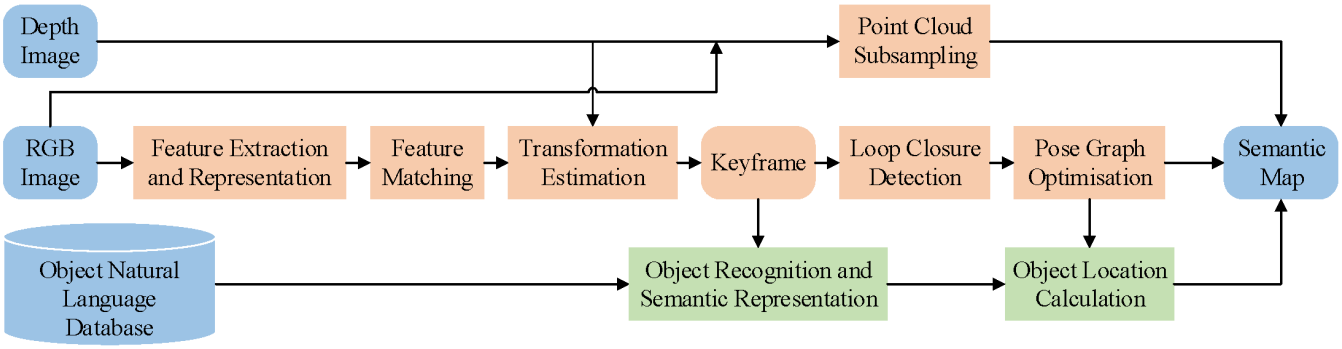


Fig. 1: Semantic mapping process. Blue boxes: inputs and output. Orange boxes: mapping process. Green boxes: semantic information extraction process.

Chen *et al.* [17] and Zhou *et al.* [18] applied CNN for place recognition. Recent results indicate that the generic descriptors extracted from CNN are versatile and transferable, i.e. even though a model is trained for object recognition, it can also be applied for place recognition or object detection [19] and often outperforms traditional hand engineered features [20].

In this paper, we adopt the Inception-v3 model proposed in [21] for object and room recognition. Both RGB and depth images are deployed for estimating sensor poses and then generating 3D geometric maps. Semantic information extraction and room recognition are carried out through RGB images by matching the recognition results with the texts in the database. The methods are detailed in the next section.

III. Methods

A. Semantic Mapping

Fig. 1 presents the block diagram and configuration of the proposed semantic mapping process. The blue blocks are the inputs (RGB images, depth images and database) and the output (semantic map) of our system. The orange ones represent the 3D mapping process and the green ones describe the semantic information extraction process.

As for the mapping process, specific features are detected and feature descriptors are computed and obtained from RGB images. The descriptors are then compared with the ones of the previous key-frame and their 3D coordinates to estimate a rough transformation matrix (rotation and translation). If the transformation is substantial enough, a new key-frame is added. Subsequently, loop closure detection is carried out by matching the current key-frame with some of the previous key-frames. A pose graph is then built and optimized using g2o [22] in order to obtain a relatively precise trajectory. Finally, the point clouds generated from the input RGB and depth images are down-sampled and projected into a common coordinate frame. A detailed 3D model is thus represented.

1) Feature Extraction: In our system, we adopt SURF (Speeded up Robust Features) [23] for both mapping and object recognition. SURF has claimed to be several times faster than SIFT, and the accuracy still remains relatively acceptable. SURF employs integral images and square-shaped filters to approximate the determinant of the Hessian matrix during Gaussian smoothing, thus can be computational efficient.

We have also tested SIFT (Scale Invariant Feature Transform) and ORB (Oriented FAST and Rotated BRIEF) features. SIFT tends to be more computationally expensive, however the Absolute Trajectory Error (ATE) remains almost the same. Original ORB feature is several times computationally efficient than SURF, but the good matches that provide correct transformation estimation (inliers) are always not enough. Therefore, we finally decided to employ SURF in our system.

2) Transformation Estimation: Due to the distinctive visual features in indoor environments, the motion of the sensor can be estimated by measuring the similarity or the distance between the feature descriptors extracted from two sequential key-frames. In this paper, we apply Perspective-n-Point (PnP) to solve this problem, which originates from camera calibration. PnP solves the problem of estimating the pose of a calibrated camera given a set of n points in the 3D world and their corresponding 2D projections in a image. The sensor motion which consists of 6 degrees of freedom (DOF) can thus be estimated and represented by a rotation matrix (roll, pitch and yaw) and a translation matrix.

$$s\boldsymbol{p} = \boldsymbol{C}\left[\begin{array}{c|c} \boldsymbol{R} & \boldsymbol{T} \end{array}\right]\boldsymbol{P} \tag{1}$$

or

$$s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix}\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{2}$$

where s is the scale factor for the image point, (u, v) are the 2D coordinates of a point in the current frame, (x, y, z) are the coordinates of its associated 3D point in the previous frame, f_x, f_y are the focal lengths expressed in pixel units, (c_x, c_y) is a principal point that is usually at the image center, $\boldsymbol{R}$, $\boldsymbol{T}$ are the expected rotation and translation matrix, respectively.

Although four pairs of points are enough to estimate the transformation, some possible errors could appear during

matching and may affect the result. Therefore, we apply RANSAC to retain good matches that provide correct estimation and improve the robustness in terms of outliers. Particularly, we minimize the sum of squared distances between two frames to yield the best transformation matrix. If the transformation is substantial enough, the current frame is regarded as a new key-frame.

3) Loop Closure Detection: A global pose graph can be generated by the transformation estimation process discussed above. However, the rough individual estimations between pairs of consecutive key-frames are noisy, especially when few features are detected. Loop closure is thus applied to reduce the accumulated noise and increase the mapping accuracy by comparing the current key-frame with the previous frames. This inevitably builds up the computational cost linearly due to the increasing number of processed frames. Although a computer with multi-core processors mitigates such problem to a certain degree, the comparison of the current key-frame to all the earlier frames is not feasible.

Moreover, revisiting the same places only occasionally occurs and a successful loop closure is not always available. Therefore, we adopt a more efficient strategy [11] to select the candidate frames. In order to reduce the number of candidate frames in our system, they are only selected from the set of key-frames. We first detect loop closure in several previous neighboring key-frames. Subsequently, several key-frames are randomly selected with a preference for much earlier ones to estimate transformation with the current key-frame. When we revisit the same scene and a loop closure is found, more key-frames are further explored in the neighboring frames of this one to find the best match. Finally the rotation and translation matrix calculated based on the least transformation distance are applied to the global pose optimization process.

4) Graph Optimization: The edges in the pose graph are generated by transformation estimation between pairs of key-frames. However, they may fail to form a globally consistent trajectory due to estimation errors. In this paper, we adopt the g2o framework [22] which performs a minimization of non-linear error function. The optimization result can be directly represented as our global trajectory. More specifically, the errors can be minimized by finding the minimum of a function of this form:

$$\boldsymbol{F}(\boldsymbol{x}) = \sum_{(i,j)\in C} e(\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{z}_{ij})^T \boldsymbol{\Omega}_{ij} e(\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{z}_{ij}) \quad (3)$$

where $\boldsymbol{x} = (\boldsymbol{x}_1^T, \cdots, \boldsymbol{x}_n^T)^T$ is the vector of sensor pose calculated by each key-frame. $\boldsymbol{z}_{ij}$ and $\boldsymbol{\Omega}_{ij}$ represent respectively the mean and the information matrix of a constraint relating the poses $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. $e(\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{z}_{ij})$ is a vector error function that measures how well the poses $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ satisfy the constraint $\boldsymbol{z}_{ij}$. The error function $e(\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{z}_{ij})$ is 0 when the estimated transformation matrix is exactly the true value.

In our system, global pose graph optimization is performed when all the key-frames are detected. We have found that graph optimization is of great value when the sensor recaptures the same scene after traveling for a long distance, since the non-linear error function substantially reduce the accumulated noise.

5) Semantic Information Extraction: This process consists of object detection and semantic representation.

The pretrained Inception-v3 model is deployed for object detection in each key-frame. To eliminate errors, we set three rules:

- The top result score has to exceed a threshold.
- Only the objects in the database can be recognized.
- An object has to be recognized at least 4 times in 4 continuous key-frames.

Once an object is detected, the key-frame with the highest score is selected. The translation matrix from the center of the image to the current key-frame is calculated as the object local coordinates. After pose graph optimization, the object location is recalculated to obtain the global coordinates.

Semantic representation is the interpretation process from objects or places to a human-compatible language. In line with the way humans categorize rooms, our conceptual hierarchy is designed by using "has-a" link [24], i.e. room types are defined on the basis of the objects they contain. A database is also created to store object names, room types, the "has-a" relations and the locations of the detected objects.

B. Room Recognition

Visually impaired people feel nervous if they are not aware of their locations. In our system, a fast and accurate result can be obtained by only two frames. Fig. 2 shows our room recognition process, which is carried out for both input images. If the results are identical, we output the result directly. If the results are different, we apply object recognition and try to detect objects in the 2nd image. If a representative object is found, the result is the associated room. If not, we choose the result with the higher score from room recognition step as the

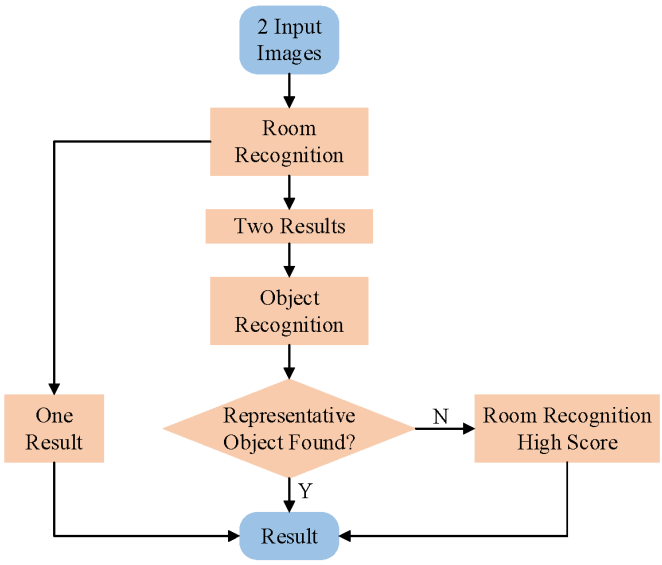
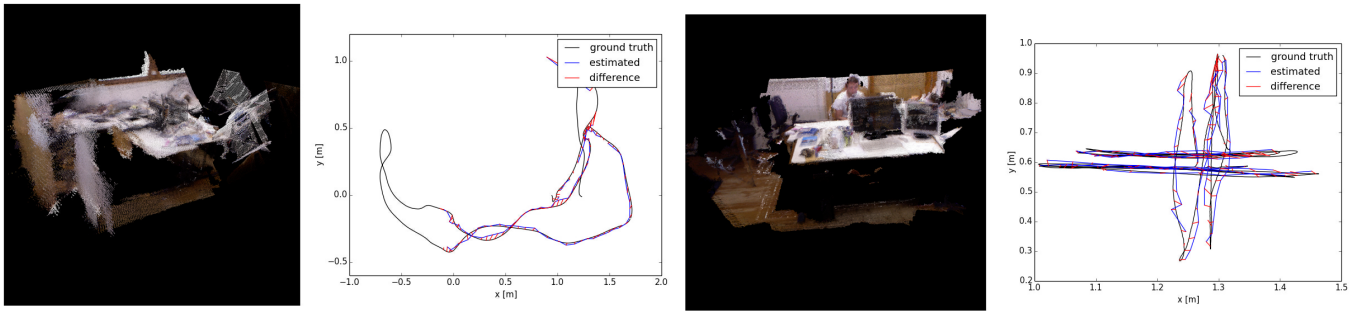
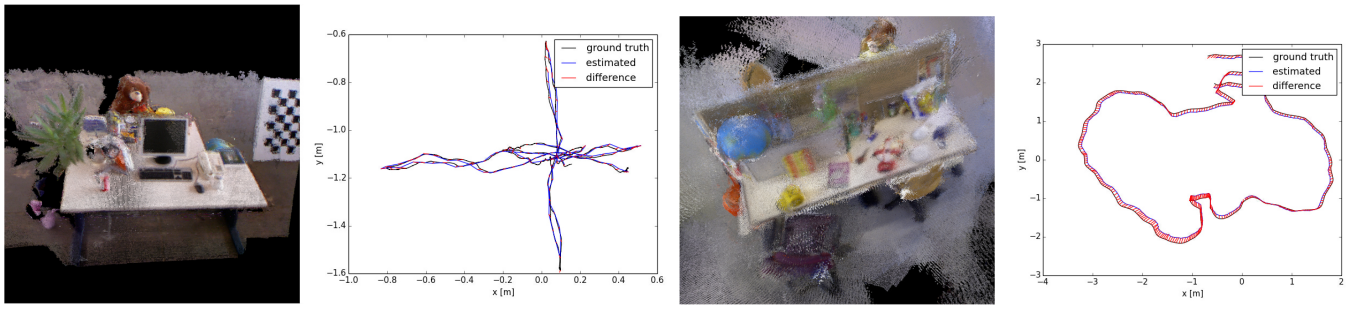


Fig. 2: Room recognition process.



(a) fr1/desk.

(b) fr1/xyz.



(c) fr2/xyz.

(d) fr3/long.

Fig. 3: 3D mapping test on the benchmark dataset.

TABLE I: Performance analysis based on the benchmark and our dataset.

Dataset	Frames	Key-frames	RMSE of ATE (m)
fr1/desk	573	119	0.064
fr1/xyz	792	153	0.013
fr2/xyz	3615	139	0.015
fr3/long	2488	447	0.028
our lab	1528	415	0.055

output. Here we consider the top three results during object recognition.

The model deployed is a Inception-v3 model [21] developed by Google company, which has already been tested by the ImageNet Large Visual Recognition Challenge using the data from 2012. The challenge is a standard task in computer vision where models try to classify the entire images into 1000 classes. The model achieved by setting a top-5 error rate of 3.46% on the 2012 validation data set. We have found that the model also has a high performance for place recognition task.

IV. Experiment Results

In this section, the semantic mapping and room recognition subsystems are both evaluated. For semantic mapping, a test in our lab is first presented. A Vicon motion tracking system is used to provide the ground truth of our sensor movement. We then test our mapping algorithm on the TUM benchmark [25] and compare our result with the RGB-D SLAM algorithm presented in [11]. Subsequently, we test our system in a home environment for room recognition. We employ a Microsoft Kinect for Xbox One to gather data in our lab and a Asus Xtion PRO LIVE in the home environment. An Intel Core i7-3632QM @ 2.2GHz CPU is used for semantic mapping experiment. An Intel Core i7-3370 @3.4GHz CPU and a GeForce GTX 980 GPU are used for room recognition experiment. For object and room recognition, we deploy TensorFlow open source software library [26].

A. Semantic Mapping

We tested our mapping algorithm on a benchmark dataset. To evaluate our mapping performance, we adopted the root-mean-square Absolute Trajectory Error (ATE) which directly calculates the deviations between pairs of estimated and ground truth poses, as shown in Table I. Both poses were preprocessed and associated using timestamps. Fig. 3 presents the 3D maps obtained and the trajectory deviations. We can see that our algorithm can track almost all of the frames and estimate a relatively accurate trajectory. We adopted the benchmark tool provided in [25] to draw a global trajectory. The black line is the ground truth, the blue line is the estimated pose graph and the red lines shows the differences.

However, we still failed to track part of the sensor motion and obtained high ATE values, e.g., left part of the fr1/desk dataset trajectory. One reason is because the sensor was facing to a plain scene, thus there were not enough features extracted from the RGB images. Another reason is that the rotation speed of the sensor was too high and the scenes in neighboring frames suffered from a substantial change, which caused the number of good matches smaller than the threshold that was used to launch the motion estimation process.

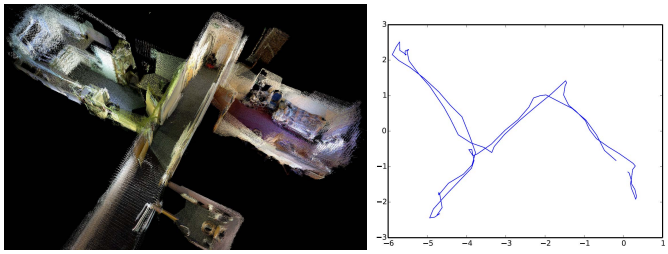
TABLE II: Performance comparison between our algorithm and the one presented in [11] based on RMSE of ATE values. (unit: metre)

Dataset	Our Algorithm	Algorithm in [11]
fr1/desk	0.064	0.026
fr1/xyz	0.013	0.014
fr2/xyz	0.015	0.008
fr3/long	0.028	0.032

Another mapping algorithm in [11] was used to compare with our results. Table II shows a comparison between our mapping subsystem and the one in [11]. It is clear that our algorithm needs further improvement when the sensor is working at a high speed.

B. Room Recognition

We carried out the room recognition experiment in a flat including a bedroom, a kitchen and a toilet. In this experiment, 7 objects were added to the database: bed, lamp, wardrobe, microwave, stove, fridge, toilet seat. A handheld Xtion sensor and a laptop were used to map the environment. CNN model training and room recognition were carried out on a desktop with GPU. The environment 3D map is shown in Fig. 4.



(a) 3D map. (b) Trajectory.

Fig. 4: 3D map of the flat.

We selected 2144, 2888, 726 images of the bedroom, kitchen and toilet for training. Some of the training images are shown in Fig. 5. The training steps were 22000 and it took 4 hours. The cross-entropy loss is shown in Fig. 6.

In order to evaluate our training result, we built two test datasets based on lighting condition (in the night with lights on) and object location changes (e.g. lamp, kettle, chair, duvet, bowl, oil bottle). Fig. 7 shows the precision (room recognition by only one image) at every 500 training steps. As we can see, the precision dropped slightly after 15000 steps due to the overfitting. Thus, the model trained at 15000 step was selected.

We then applied room recognition with representative objects as stated in Section III. Table III summaries our room recognition performance. As we can see, the recognition precision was raised slightly by verifying the results with the representative objects detected in the images. We placed two images in one batch and fed it into the GPU. Therefore,

Fig. 5: Some of the selected images.

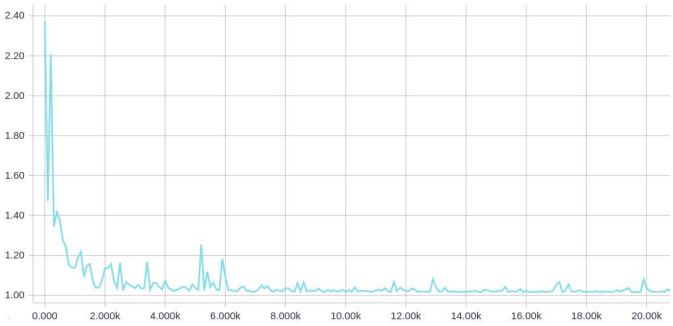


Fig. 6: Raw cross-entropy loss.

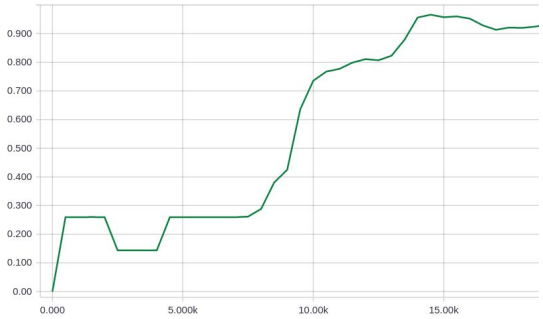
the processing time remained almost the same if the results were identical. In the case that the results were different, the processing time rose due to object detection in the images. But it is still less than 0.02s and feasible for real-time application.

V. Conclusion

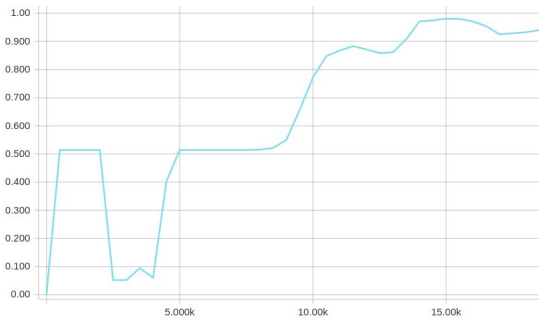
In this paper, a semantic mapping and room recognition method was presented to help visually impaired people navigate indoor environments. The system consists of a 3D camera and a computer. A simple approach was introduced to eliminating room recognition errors based on representative

TABLE III: Room recognition performance.

	Lighting Condition Change	Object Location Change
Precision without Representative Object	0.9660	0.9808
Processing Time without Representative Object	0.0088s	0.0088s
Precision with Representative Object	0.9780	0.9864
Processing Time with Representative Object	0.0183s	0.0183s



(a) Lighting change.



(b) Object location change.

Fig. 7: Training result evaluation at different training steps.

objects. The pose estimation accuracy was tested and compared with a benchmark dataset. Finally, the performance of room recognition was verified in a home environment.

Our future research will continue the development of efficient and reliable localization algorithms so that a blind person can accurately locate an object within a room by using the developed visual guide system more easily. Moreover, convolutional neural networks will be deployed to achieve more accurate visual localization results.

ACKNOWLEDGMENTS

Qiang Liu and Ruihao Li are financially supported by China Scholarship Council and the University of Essex Scholarship for their PhD studies. Our thanks also go to Robin Dowling and Ian Dukes for their technical support.

REFERENCES

[1] The World Health Organization - Visual impairment and blindness. [Online]. Available: http://www.who.int/mediacentre/factsheets/fs282/en/

[2] D. F. Wolf and G. S. Sukhatme, "Semantic mapping using mobile robots," *IEEE Transactions on Robotics*, vol. 24, no. 2, pp. 245–258, 2008.

[3] Q. Liu, R. Li, H. Hu, and D. Gu, "Extracting semantic information from visual data: A survey," *Robotics*, vol. 5, no. 1, p. 8, 2016.

[4] A. Nüchter and J. Hertzberg, "Towards semantic maps for mobile robots," *Robotics and Autonomous Systems*, vol. 56, no. 11, pp. 915–926, 2008.

[5] A. Flint, C. Mei, D. Murray, and I. Reid, "A dynamic programming approach to reconstructing building interiors," in *11th European Conference on Computer Vision (ECCV)*. Springer, September 2010, pp. 394–407.

[6] Y. Tian, X. Yang, and A. Arditi, "Computer vision-based door detection for accessibility of unfamiliar environments to blind persons," in *12th International Conference on Computers Helping People with Special Needs (ICCHP)*. Springer, July 2010, pp. 263–270.

[7] J. Wu, H. Christensen, J. M. Rehg *et al.*, "Visual place categorization: Problem, dataset, and algorithm," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, October 2009, pp. 4763–4770.

[8] F. Neves dos Santos, P. Costa *et al.*, "A visual place recognition procedure with a Markov chain based filter," in *2014 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*. IEEE, May 2014, pp. 333–338.

[9] J. Civera, D. Gálvez-López, L. Riazuelo, J. D. Tardós, and J. Montiel, "Towards semantic SLAM using a monocular camera," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, September 2011, pp. 1277–1284.

[10] L. Riazuelo, M. Tenorth, D. Di Marco, M. Salas, D. Gálvez-López, L. Mosenlechner, L. Kunze, M. Beetz, J. D. Tardos, L. Montano *et al.*, "RoboEarth semantic mapping: A cloud enabled knowledge-based approach," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 2, pp. 432–443, 2015.

[11] F. Endres, J. Hess, J. Sturm, D. Cremers, and W. Burgard, "3-D mapping with an RGB-D camera," *IEEE Transactions on Robotics*, vol. 30, no. 1, pp. 177–187, 2014.

[12] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, "KinectFusion: Real-time dense surface mapping and tracking," in *2011 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, October 2011, pp. 127–136.

[13] J. Stückler, B. Waldvogel, H. Schulz, and S. Behnke, "Dense real-time mapping of object-class semantics from RGB-D video," *Journal of Real-Time Image Processing*, vol. 10, no. 4, pp. 599–609, 2015.

[14] A. Hermans, G. Floros, and B. Leibe, "Dense 3D semantic mapping of indoor scenes from RGB-D images," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 2631–2638.

[15] R. B. Rusu, "Semantic 3D object maps for everyday manipulation in human living environments," *KI-Künstliche Intelligenz*, vol. 24, no. 4, pp. 345–348, 2010.

[16] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014, pp. 580–587.

[17] Z. Chen, O. Lam, A. Jacobson, and M. Milford, "Convolutional Neural Network-based place recognition," *arXiv preprint arXiv:1411.1509*, 2014.

[18] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning deep features for scene recognition using places database," in *Advances in Neural Information Processing Systems*, 2014, pp. 487–495.

[19] A. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: an astounding baseline for recognition," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2014, pp. 806–813.

[20] N. Sunderhauf, S. Shirazi, F. Dayoub, B. Upcroft, and M. Milford, "On the performance of ConvNet features for place recognition," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 4297–4304.

[21] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception architecture for computer vision," *arXiv preprint arXiv:1512.00567*, 2015.

[22] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g 2 o: A general framework for graph optimization," in *2011 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, May 2011, pp. 3607–3613.

[23] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded Up Robust Features," in *9th European Conference on Computer Vision (ECCV)*. Springer, May 2006, pp. 404–417.

[24] H. Zender, O. M. Mozos, P. Jensfelt, G.-J. Kruijff, and W. Burgard, "Conceptual spatial representations for indoor mobile robots," *Robotics and Autonomous Systems*, vol. 56, no. 6, pp. 493–502, 2008.

[25] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, October 2012, pp. 573–580.

[26] TensorFlow. [Online]. Available: https://www.tensorflow.org/