Computing a Perceptual Map Using a Stereo-Vision Mobile Robot

S. K. Eng[1], B. E. Khoo[1] and W. K. Yeap[2]
[1]*School of Electrical and Electronic Engineering, Universiti Sains Malaysia, Malaysia.*
[2]*Centre for Artificial Intelligence Research, Auckland University of Technology, New Zealand.*
khengkent@gmail.com

***Abstract*—A new computational model of how humans integrate successive "local environments" obtained as views at limiting points in the environment to create a perceptual map has been proposed and validated using a laser-ranging mobile robot. Compared with the SLAM (Simultaneous Localization and Mapping)-based approach, the proposed process is less computationally demanding and provides an interesting account of how humans compute their cognitive maps. Since vision plays an important role in how humans compute their maps, we extend the previous work by implementing the model using a vision-based mobile robot. Specifically, our model takes a pre-recorded series of stereo-vision images of a large indoor environment at USM and produces a perceptual map. The results show that the model is not dependent on the use of a laser-ranging device and this is significant if the model is intended as a cognitive model of spatial cognition.**

***Index Terms*—Human Perceptual Map; Mobile Robot; Stereo-Vision Images; SLAM.**

I. INTRODUCTION

When an autonomous mobile robot moves in an unknown environment, it needs to incrementally build a map of the explored environment and uses the map to estimate its location in the environment. This is the key issue addressed in the Simultaneous Localization And Mapping (SLAM) [1] approach. The main goal of SLAM then is to create an accurate map of the environment. The SLAM problem has gained significant attention from robotics researchers over the years [1,2]. Laser-based systems [3] and the vision-based systems [4] are the two most popular sensors for robotic researchers using the SLAM-based approach.

While the SLAM-based methods have been shown to perform well in producing an accurate map of the environment, it is interesting that human (and animals) do not compute such a map [5]. Instead, based on studies by researchers interested in cognitive mapping, it has been shown that human memory pertaining to the environment is incomplete, inaccurate (in the metric term) and fragmented [6-8]. How do they compute such a map? The process, according to these researchers, is an intriguing one. It appears that what is learned could be affected by seemingly unrelated factors such as the direction of travel and presence of barriers. Subjects also show that they learn both a topological map and a metric map and the latter shows local accuracy (i.e. some parts are accurate while others are either missing or inaccurate). Consequently, there has been much debate as to the nature of the process and the map computed.

In [9], it is argued that despite the controversies surrounding the learning of a cognitive map of one's environment, what is computed initially ought to be a map of the environment that is incomplete and inexact. This map is referred to as a perceptual map, a map obtained via one's direct experience of the environment. The process suggested is intriguing: it requires no updating and no error correction. All it needs is the tracking of familiar objects in view. It is a parsimonious process for learning a map of the environment. The model is tested on a mobile robot equipped with a laser sensor and an odometer. In this paper, we implement the model on a mobile robot equipped with stereo vision. Specifically, stereo-vision images of a large indoor environment at USM are captured, and computer vision and mapping algorithms for the map building process are employed to compute a perceptual map of the environment.

Testing the model using vision is essential for two reasons. First, vision plays a vital role in the development of human spatial cognition [10] and as such our work here is to test the model in [9] using vision. However, note that existing computer vision is unlike human vision. For example, the latter is extremely illusory as it does not provide a true geometrical description of the environment in view. It provides a rich source of information but it has high visual acuity only in the small foveal region of the retina, and thus a large part of the input lacks clarity and detail. The eyes need to make rapid movements (known as saccades) to bring different regions into the foveal. Consequently, in our implementation, there is no attempt to simulate computer vision. Our goal is to replicate that of [9] using a simple vision system to investigate if one could still compute a reasonable map that is inexact and incomplete. If successful, more complex visual systems could be introduced later. Second, computer vision systems are less expensive and are gaining popularity as a sensor for robotic systems and in particular for drones. Developing such a process for robot use would be an attractive alternative to the SLAM-based approach.

The remainder of this paper is organised as follows. In section II, the principle of human perceptual mapping is presented. The methodology used in the experimental study is described in section III, while the results and discussion are presented in section IV. Conclusions and suggestions for further work are given in section V.

II. THE HUMAN PERCEPTUAL MAP BUILDING ALGORITHM

A perceptual map is defined as a global metric map of one's environment that is the result of one's direct experience of the environment. In [9], two assumptions are made. First, a view affords one a map of a local environment that one is about to enter. Second, the world we live in is relatively stable. Consequently, to compute a perceptual map, one goes into a

cycle: get a view which shows a map of the local environment that one is about to explore, explore inside the local environment, if one moves out of the local environment, repeat. Note that the second assumption means that we do not need to update the view at all while exploring the local environment. However, if so, how does one know where one is in the environment? This is achieved by tracking familiar objects in every subsequent view [9]. The familiar objects identified in the environment are known as the reference objects. When one moves out of the current local environment, one remembers another view and enters into another local environment. A list of such views forms a perceptual map. If we allow these views to have an overlapping region containing some common objects (Figure 1), then we can create a single global map that is inexact and incomplete.

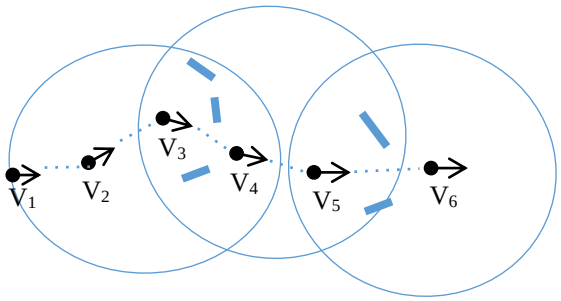

Figure 1: Creating a global perceptual map: V_1..V_6 are six robot viewing positions, and the circles indicate the view boundary for V_1, V_3 and V_5. The perceptual map consists of V_1, V_3 and V_5. By triangulating the viewing position of V_5 in V_3 and then V_3 in V_1 using the common surfaces (solid line) respectively, one could create a global map of the three views.

The perceptual map building algorithm implemented in a mobile robot [9] is described formally as follows. Let V_0 be one's starting view, R be a series of reference objects detected in V_0, and PM is the perceptual map. Initialize PM with V_0. For each movement of the robot in the environment, perform:

i. Execute the "move" command and obtain a new view, V_n;
ii. Identify the tracked reference objects in V_n by transforming the previous view to the current view, i.e., tracking;
iii. If the number of tracked reference objects found are fewer than a pre-specified number, use V_{n-1} to expand PM and V_{n-1}, to create a new R;
iv. Remove the tracked reference objects in R that have disappeared in the current view. Go to step (i).

The theory leaves open three implementation issues, namely; (i) how and what reference targets should be selected for tracking purposes in successive views; (ii) when the perceptual map needs to be expanded (i.e. step (iii) above); and (iii) the mechanism for expanding information in the perceptual map (i.e. step (iv) above).

III. Methodology

In this paper, instead of using a laser-ranging mobile robot, a stereo-vision mobile robot (as shown in Figure 2) is used to compute a perceptual map. Specifically, stereo-vision images of the environment are used as input to compute a perceptual map of the environment. Details of the proposed perceptual map building procedure using stereo images and robot's displacement information (rotation and translation) are explained, as follows.

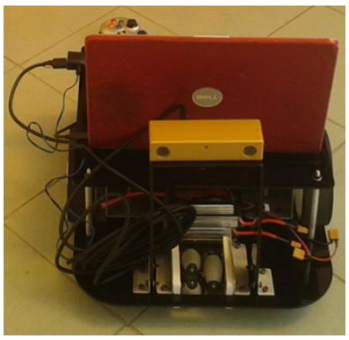

Figure 2: The platform of the vision-based mobile robot.

A. Stereo-vision Images

In this research, the images are captured using a stereo vision-based mobile robot. The PointGrey BumbleBee stereo camera [11] is used, and its configuration is as shown in Table 1.

Table 1
The Configuration of the PointGrey BumbleBee Stereo Camera

Parameters	Configurations
Baseline (b)	11.9952 cm
Focal length (f)	251.48735 pixel
Principal Point coordinates (c_0, r_0)	(240/2, 320/2) pixel
Image size	320 x240 pixel

1) Preprocessing Step

When the robot moves from one position to another, its odometry provides information of the robot's displacement (rotation and translation) relative to the starting position. However, here, the odometry information is retrieved from analysing the associated camera images using our proposed algorithm. In this case, subpixel locations of the edge points in the previous and the current views are detected using the method as proposed in [12]. Then, the features of edge points in the previous view with those in the current view are matched to retrieve the corresponding edge points of both views. After that, the weak matched pairs are removed. At the same time, the x and z location of each corresponding edge points from both views in cm are retrieved using:

$$z=\frac{bf}{d} \tag{1}$$

$$x=(c-c_0)\frac{b}{d} \tag{2}$$

where b and f are the baselines and focal length of the stereo camera, respectively, c and c_0 are the column(x) coordinates of the edge points and the column(x) principal point in the right reference image (in pixel values), and d denotes the disparity value.

The next crucial step is to infer the spatial transformation from two selected corresponding control edge points. The output returns a 3x3 spatial transformation matrix, which contains information of the robot's displacement (rotation (angle) and translation(tx,tz)) relative to its last step.

2) Reference Objects

Reference objects (specifically edges of the reference objects in this research) are essential for representing the spatial information of the environment in the perceived scene. These objects help the viewers to locate their positions in the environment. In this research, subpixel location of the edge points in the right and left images are detected using the method in [12]. Then, the features of edge points in the right image are matched with that in the left image, so that the corresponding edge points in them can be obtained. The weak matched pairs are removed, and the disparity values for each edge point in the right image can be obtained. The locations of each edge points (in cm) in the right image are retrieved by using the Equations 1 and 2. A filtering process is also performed to remove the edge points that are too far away (i.e. 2000cm, where the accuracy of the distance measurement from the camera becomes low) from the robot location. Figure 3 shows the final edge points of the image.

The next step is to cluster all detected reference points (in cm) using a grid-based method. In this case, the size (width and height) of the grid area is first determined. The reference points that fall into the same grid area are considered as belonging to the same cluster and is represented by one reference object. Figure 4 illustrates an example of the result.

The last step is to use a 2-point line to represent every referenced object (a group of reference points) in each occupied grid area. In this case, the slope(m) and offset(b) values of each group of reference points are first calculated by using a linear regression method. Secondly, identify the minimum x position value (x_{p1}) and the maximum x position value (x_{p2}) of each group of reference points, and finally, the 2-point line (x_{p1} z_{p1}; x_{p2} z_{p2}) for representing every referenced object can be created by using the equation shown as follows;

$$z_{p1}=m*\mathrm{x}_{\mathrm{p1}}+b \tag{3}$$

$$z_{p2}=m*x_{p2}+b \tag{4}$$

The reference objects (line surfaces) with the minimal size of length (i.e., smaller than 5cm) are removed. An example of the result is shown in Figure 5. From Figure 5, every referenced object is assigned an ID (identification) number.

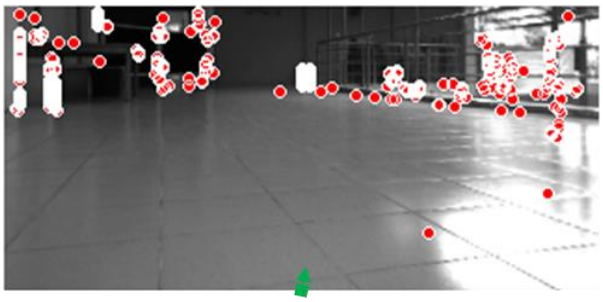

Figure 3: The edge points in an image. Green box denotes the robot position, and the green arrow denotes the robot orientation.

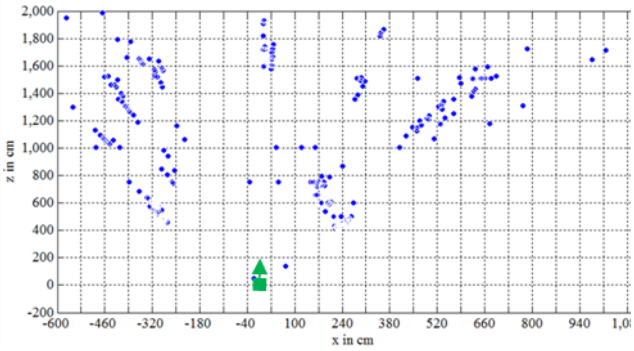

Figure 4: Clustering the reference points using a grid-based method. Green box denotes the robot position, and the green arrow denotes the robot orientation.

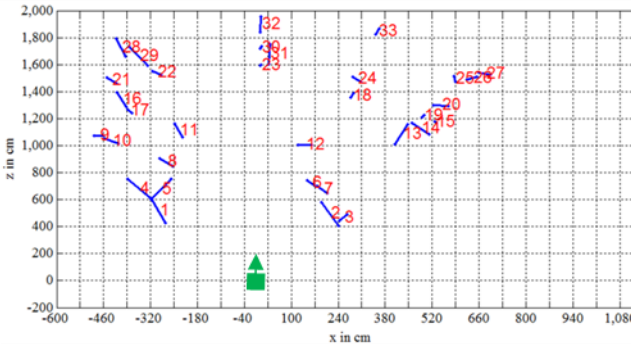

Figure 5: Reference objects with their respective ID (identification) numbers. Green box denotes the robot position, and the green arrow denotes the robot orientation.

B. Recognizing and Tracking Reference Objects

This step aims to recognise and track the reference objects identified in the first view (i.e. those lines with an ID number in Figure 5). Firstly, the reference objects in the current view are identified. Then, the locations of the tracked reference objects in the previous view are projected based on the coordinate system of the current view using:

$$[x'_{pi}\ \ z'_{pi}\ \ 1] = [x_{pi}\ \ z_{pi}\ \ 1]* \text{ spatial transformation matrix} \tag{5}$$

where pi denotes the first point and the second/endpoint in the reference objects/line surfaces. (x_{pi} , z_{pi}) is the location of each point in the reference objects/line surfaces (each line consists of two points) in the previous view (of the previous view coordinate system); (x'_{pi} , z'_{pi}) is the transformed location of each point in the tracked reference objects /line surfaces in the previous view (of the current view coordinate system). The 3x3 spatial transformation matrix, which consists of the displacement (tx, tz and angle) of the robot relative to the last step is retrieved from the pre-possessing step.

There are now two copies of the reference objects in the current view – one consists of reference objects initially found in the current view (RO_B) and another consist of the transformed tracked reference objects from the previous view (TRO_{AtoB}). Those reference objects in the new view (i.e., RO_B) that are near the transformed locations of the reference objects (i.e., TRO_{AtoB}) are considered as the remaining reference objects (also known as tracked reference objects).

C. Expanding the Map

When the robot detects fewer than a pre-specified minimum number of tracked reference objects in the current view, it realises that it has now moved out of the current local environment. The robot adds its previous view to the global map, because it may not be able to triangulate its position using the current view. One of the essential tasks is to compute the spatial location of each new reference object (line surface) relative to the nearest common tracked reference object (i.e., L^x_{pv}). The location of the nearby reference objects can be encoded as a vector with its length equal to the distance from the selected end-point and its angle equal to the angular displacement from the surface slope (as illustrated in Figure 6). The procedure for this step is as follows.

i. Retrieve the common tracked reference objects in the initial view (CRO_0) and the previous view(CRO_1);
ii. Identify the nearest common tracked reference object (CRO_1) for each new reference object in the previous view. The new reference objects are to be added to the global map;
iii. Compute the vector and angular displacement (i.e., v_i and Θ_i), of each new reference object from the left end-point of the nearest common reference object (i.e. L^x_{pv}) in the previous view;
iv. Compute the position of each new reference object in the global map by using the left end-point of the same common reference object found in the initial view (i.e., L^x_{gm}) , with its corresponding vector length (i.e., v_i) and new computed angle ($\Theta_{i\_new} = \Theta_i - \Theta_{turn}$), where Θ_{turn} represents the accumulated turn-angle of the robot;
v. All the new reference objects in the previous view are added to the global map. Note that the global map is first initialised using the first view.

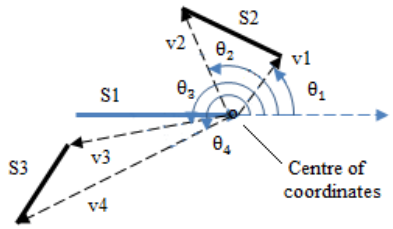

Figure 6: Computing the spatial locations of reference objects (i.e., S1 and S2) close to a common tracked reference object. S1 is recognised as a common tracked reference object and S2 and S3 are coded using two pairs of vectors centred on the right end-point of S1[13].

D. Remove Redundancy points in the Global Map for Boundary Computation

This step aims to remove redundant points in the global map, in order to obtain a clean map. Through this process, we can visually judge whether the shape of the global map corresponds to the actual environment. Besides that, the spatial information of the environment can be clearly observed when the boundary of the global map is created. It is accomplished by first removing the reference objects which intersect with the path line. Then, the inner and outer reference objects are identified. Figure 7 shows an example of the outcome. For each path line, select an appropriate number of important inner and outer reference objects that have the nearest distance with the current path line point. Many redundant reference objects in the global map can be removed through this process. To form the boundary, determine the initial reference object. Then, connect it from the current point to the next point accordingly.

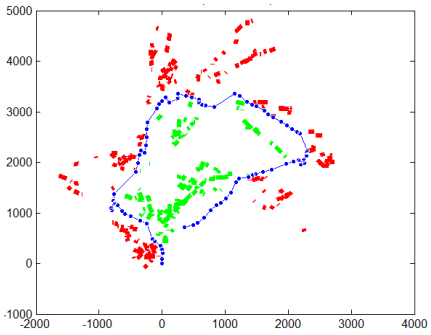

Figure 7: The initial global map with the identified inner (green line) and outer (red line) reference objects. The blue lines indicate the path lines and the blue circles indicate the path points.

IV. Results and Discussion

In this paper, the perceptual map model [9] is implemented and evaluated using a vision-based mobile robot with respect to an environment at the first floor of the Lecture Hall Complex of the Engineering Campus of Universiti Sains Malaysia.

During the evaluation, the robot explores the environment in both clockwise and anti-clockwise directions. The goal of the experiments is to examine whether inaccurate and incomplete global maps can be computed using the proposed procedure in Section III. This means that the global maps computed should be able to represent the overall spatial information of the environment traversed [5,9,14].

It is important to emphasise that the model proposed in [9] does not attempt to correct any sensor errors since it is not necessary to continuously update the sensor readings when the perceptual map is built. What is added as the new local environment is precisely what is seen in the previous view. The rationale is that the environmental details are not necessary as long as the overall spatial information of the environment is recognised [6]. The purpose of exploring the environment in both clockwise and anti-clockwise (reverse) directions is to make sure that the maps computed are not direction dependent.

In the experiment, the robot explores an indoor environment with a size of 28m x 28.5m in a clockwise and anti-clockwise direction. Figure 8 to Figure 11 show respectively the floor plan, the initial global map, a global map with redundant objects removed, and global map with boundary computed for the indoor environment. Note that S and E denote the starting point and end point of the robot, respectively. The horizontal line denotes the x position in cm, and vertical line denotes the z position in cm.

The initial location and orientation of the robot are the same for the clockwise and anti-clockwise directions. From Figure 9(a), it is observed that a group of tracked reference objects near location A (refer to its corresponding floor plan) is clearly shown. From Figure 9(b), a group of tracked reference objects near location B is also clearly shown. On the other hand, the path lines (obtained from odometry information) computed for Figure 9(a) are accurate (refer to the path lines in corresponding floor plan), but specific path lines in Figure 9(b) (as highlighted in circle) is less accurate. This is because of the odometry error. In addition, the size of the map in Figure 9(a) and Figure 9(b) appears almost the

same; it is because of the orientation of the robot in starting points are the same.

From Figure 10, it can be observed that the important reference objects nearer to the path line are retained, while other redundant objects (i.e., located outside the important reference objects) are removed. From Figure 11(b), it is shown that the highlighted (in circle) boundary of the green line is less correct, due to less correct path lines. Nevertheless, the shape of the map is still intact. Overall, the proposed procedure can compute a map that depicts a reasonably accurate shape of the environment from both clockwise and anti-clockwise directions, even in the presence of minor odometry errors in the anti-clockwise direction.

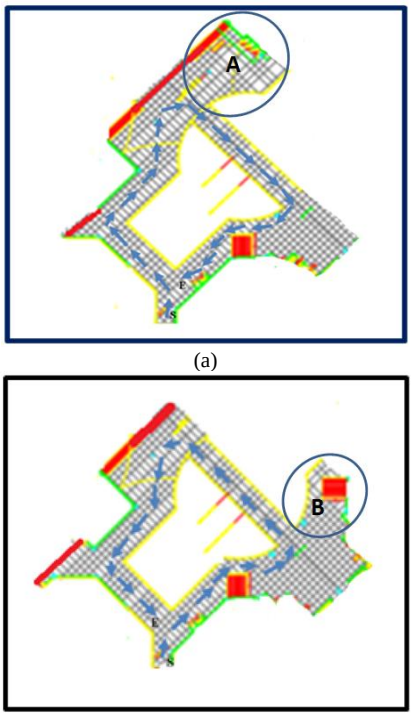

Figure 8: Floor plan in half route environment (a) with clock-wise direction (b) with anti-clockwise direction.

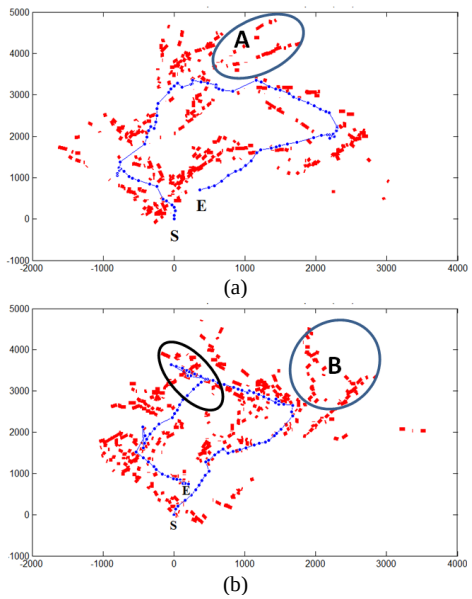

Figure 9: Initial global map in half route environment. The blue line and red line indicate the path line and the surfaces. (a) with clock-wise direction, (b) with anti-clockwise direction.

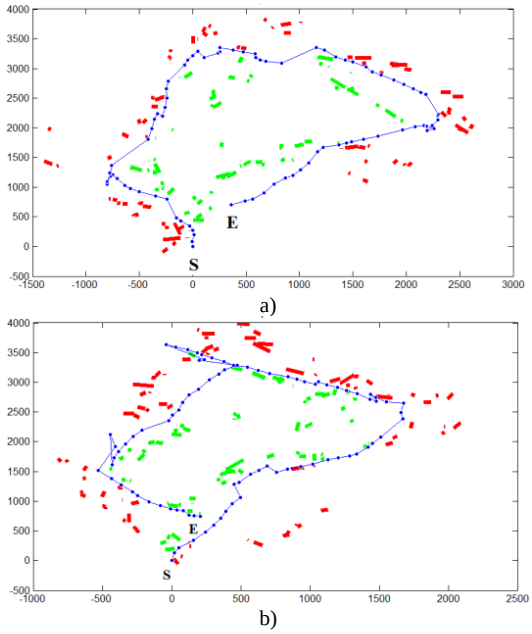

Figure 10: Global map with redundancy objects removed in half route environment. The blue line, red line and the green line indicate the path line, inner surfaces and outer surfaces. a) with clock-wise direction b) with anti-clockwise direction.

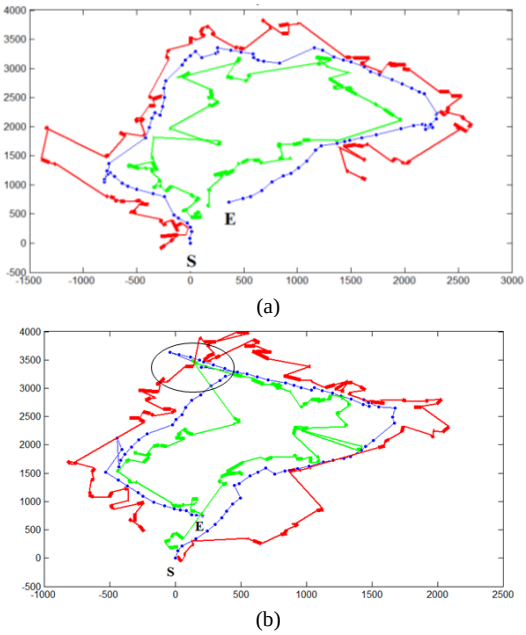

Figure 11: Boundary computed for the global map in half route environment. The blue line, red line and the green line indicate the path line, inner surfaces and outer surfaces. (a) with clock-wise direction, (b) with anti-clockwise direction.

V. CONCLUSION

In this research, computer vision and mapping algorithms for building a perceptual map based on the model in [9] have been successfully developed. The map building procedure has been evaluated using a stereo-vision mobile robot in a large indoor environment. The results indicate that the model is not dependent on the use of a laser-ranging device. The algorithms can create imprecise and incomplete maps with a good match between the created and actual spatial shape of the environment.

Further work will focus on developing more robust algorithms for vision-based robot mapping. Comprehensive evaluations of the algorithms with more complicated indoor and outdoor environments will also be conducted.

ACKNOWLEDGEMENT

The financial support of the Fundamental Research Grant Scheme (No. 203/PLECT/6711229), Ministry of Education Malaysia (MOE), UniMAP, and USM for this research is highly appreciated.

REFERENCES

[1] G. Dissanayake, S. Huang, Z. Wang, and R. Ranasinghe, "A review of recent developments in Simultaneous Localization and Mapping," in *6th IEEE international Conference on industrial and information System*, 2011, pp. 477-482.

[2] A. Gil, O. M. Mozos, M. Ballesta, and O. Reinoso, "A comparative evaluation of interest point detectors and local descriptors for visual slam," *Machine Vision and Applications*, vol. 21, pp. 905-920, 2009.

[3] T. Chong, X. Tang, C. Leng, M. Yogeswaran, O. Ng, and Y. Chong, "Sensor Technologies and Simultaneous Localization and Mapping (SLAM)," in *IEEE International Symposium on Robotics and Intelligent Sensors (IRIS 2015)*, 2015, pp. 174-179.

[4] H. Lim, J. Lim, and H. J. Kim, "Real-time 6-dof monocular visual SLAM in a large-scale environment," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 1532-1539.

[5] W. K. Yeap, M. Z. Hossain, and T. Brunner, "On the Implementation of a Theory of Perceptual Mapping," in *Proceedings of the 24th international conference on Advances in Artificial Intelligence*, 2011, pp. 739-748.

[6] R. M. Downs and D. Stea, "*Image and environment: Cognitive mapping and spatial behavior*," Chicago: Aldine., 1973.

[7] G. W. Evans, "Environmental cognition," *Psychological bulletin*, vol. 88, p. 259, 1980.

[8] K. Lynch, *The image of the city* vol. 1: MIT press, 1960.

[9] W. K. Yeap, "A computational theory of human perceptual mapping," in *Proceeding s of the Cognitive Science. Boston, USA*, 2011, pp. 429-434.

[10] C. Thinus-Blanc and F. Gaunet, "Representation of space in blind persons: vision as a spatial sense?," *Psychological bulletin*, vol. 121, p. 20, 1997.

[11] PointGrey, "Bumblebee Stereo Vision Camera Systems," in *BB2-03S2C-38 (datasheet)*, 2017.

[12] A. Trujillo-Pino, K. Krissian, M. AlemaN-Flores, and D. Santana-Cedres, "Accurate subpixel edge location based on partial area effect," *Image and Vision Computing*, vol. 31, pp. 72-90, 2013.

[13] Z. H. Azizul Hasan, "Robot mapping without a precise map," PhD Thesis, Auckland University of Technology, 2013.

[14] M. Z. Hossain, "How Albot1 Computes Its Perceptual Map," PhD Thesis, Auckland University of Technology, 2014.