PROBABILISTICALLY VOXELATED THREE DIMENSIONAL GRID MAP-BUILDING FOR EXPLORATION

Hafiz bin Iman1, Nahrul Khair bin Alang Md Rashid2 , Wahju Sediono3, Md. Raisuddin Khan4

1, 3, 4Department of Mechatronics Engineering, IIUM, Jalan Gombak, Kuala Lumpur, Malaysia, [1hafiz.ghazman@gmail.com](mailto:1hafiz.ghazman@gmail.com), [3wsediono@iium.edu.my](mailto:3wsediono@iium.edu.my), 4raisuddin@iium.edu.my

2 Sekolah Kejuruteraan Kimia dan Kejuruteraan Tenaga, Fakulti Kejuruteraan, Universiti Teknologi Malaysia, Johor Bharu [nahrulkhair@utm.my](mailto:nahrul@iium.edu.my)

**ABSTRACT**

With the increase of robot chassis mobility, an autonomous robot has unprecedented opportunity to explore an environment in situ. However, the mobility of the robot is hindered by the limited availability of three dimensional probabilistic model of the environment. In so doing, this research proposed and investigated the potential of using probabilistic voxelated three dimensional grid map that can produce three dimensional map probabilistically by incorporating stochastic nature of sensor reading and robot movements. The map are modelled to give direct probability of an occupied space. The grid cell are voxelized and embedded with relax logit function to emanate probability value of an occupied space. The performance of the probabilistic voxelated three dimensional grid map was tested by using scans collected a priori. These scans act as a kernel to the registration technique. Two separate sets of map were reconstructed using pose estimate obtained from Bayesian filters. The result was compared between maps produced under recursive Bayesian filter and Bayes’ rule filtering with referenced to the geometric information of the controlled environment. The probabilistic voxelated three dimensional grid map restored the geometric information of the environment with accuracy of 0.87. The runtime of the method converged to 0.23 s after fifth map registration. The method also compressed the metrical data from Hokuyo UTM-30LX by two decades in logarithmic scale. By using continuous probability value to represent occlusion, the map has the potential to be used with any Bayesian filtering family. The potential of the probabilistic voxelated grid map to be used as path planning method was also observed since the probability value in each grid cell of the map behaves similarly to vector field.

**Keywords**: SLAM, Voxelation, Probabilistic model, LIDAR, Hokuyo UTM 30LX, Probabilistic Mapping, Probabilistic map-building

INTRODUCTION

Map building is one half of a solution to simultaneous localization and mapping (SLAM). SLAM acts as a utility to solve robotic exploration in an unknown environment. SLAM is stochastic and involves error in data reading. Stochastic information from multiple sensors are used to form information gain through filtering and data fusion. Map building requires the same data fusion for local map registration. The registration method involves filtering technique to transform local map into global map. However, most of the solution for three dimensional map-building are in the form of point clouds. Point clouds involves inefficient representation of geometric data for autonomous robots [1],[2],[3][4]. These point clouds are discrete series of metrical data that results in inefficient path planning and exploration strategy. A map suiting the robotic ability to move in three dimensional space requires three dimensional map that emphasize the geometric information of an environment. In tandem to automatic motion, this map should represent the probabilistic nature of the sensor and the locomotion of the robot to be effectively used in SLAM solution. In [5], a three dimensional grid map was suggested for autonomous robot with six degree of freedom mobility. The three dimensional grid map uses grid cells seeded with binary values that represent occupancy in space. The grid cells are stacked continuously in series of cubic primitives. The binary values does not, however, reflect the nature of probabilistic model of the sensors and locomotion. Therefore, probabilistic solution of three dimensional map that incorporate both the probabilistic nature of sensors reading and uncertainty in locomotion is needed. Probabilistically voxelated three dimensional grid map are presented in this research paper to cater such need.

RELATED WORKS ON EVOLUTION OF MAP BUILDING

A confident map is instrumental for exact localization. Since mid-1990's the mobility of robots have increased from two degree of freedom to six degree of freedoms. This mobility requires map that represent an environment in three dimensional rendition. Several maps are functionally introduced in robotics which leads to the development of probabilistically voxelated three dimensional map-building technique. These maps are classified functionally as point clouds, graph maps, two dimensional occupancy grid maps, and three dimensional occupancy grid map.

Point clouds are a three dimensional depiction of spatial space using metrical value from a sensor. Point clouds often use iterative closest point matching (ICP) or scan matching for registration. Point clouds are limited by its vast information containing excessive metrical data that put computational strain to an algorithm [6], [7].

Graph maps uses graph theory to represent the environment. The concept of graph maps relates poses and features as nodes and abstracting raw sensor data into pose-graph [8]. A pose of a robot is positional information that reflects the degree of freedom of a robot. Graph maps lacks geometric structure and use mostly to optimize localization processes.

Two dimensional occupancy grid maps are probabilistic voxelation of point clouds based on metrical data of a noisy and incomplete sensor reading. In papers [9] and [10], grid map was first introduced. Two dimensional grid map uses planar primitives to represent occupancy in a plane. This method is still in use today to develop a more advance mapping and localization method [11], [12]. SLAM are often subjected to multiple scans of the environment. As autonomous robot moves in an unknown environment, a collection of maps, in the form of scans, are introduced to its system. The aggregation of these maps are call registration. This method of registration of local maps to a global map are done post-exploration explained in [13].

The branching of two dimensional grid map to three dimensional map retrofit robots with the ability to move in six degree of freedom. Octomap utilize grid map-building into three dimensional space [5], [14]. The grid cells in Octomap are binary values of probability of an occupancy in space.

This paper extend the definition of map explained in [5], [14], by using continuous probability value instead of binary. Each grid cells of probabilistically voxelated grid cell is based on the value of the probabilistic model of the sensor.

MATHEMATICAL CONCEPT OF SENSOR, LOCOMOTION AND MAP FOR PROBABILISTIC VOXELATION

The map-building technique incorporated in this paper uses probability and statistic method. The probabilistic model of the sensor is used to find the variance of the pose estimate of the robot. This variance is used to represent the voxelation of the grid cell in the map. The locomotion model based on Bayesian rule is used to help in registering the grid cell map into global map. Each grid cells is superseded with continuous value of probability using relaxed logit function. The grid cells in this research paper are identified henceforth as voxels.

In this research, the sensor used to collect the scan of the environment is Hokuyo UTM-30LX that produce planar metrical data. Hokuyo UTM 30LX output metrical data in the range of 200 mm to 60000 mm. According to the datasheet of the laser, provided by the manufacturer, Hokuyo UTM 30LX is subjected to 30 mm of sensitivity. But for the range between 1000 mm to 10000 mm, the sensitivity of the laser is less than 10 mm. This value is used to parameterize the Gaussian of reading coming from the laser: the standard deviation, σ, of the laser range is considered to be 10 mm and mean, μ, of the sensor is considered to be zero for modelling the voxel.

A recursive Bayesian filtering (RBF) technique is used as a kernel that outputs the variance and mean of the pose estimation of the position of the UTM-30LX. The variance value is used to model the voxel of the map and the mean of the pose estimate is used as an input to register the voxelation map into a global map. The recursive Bayesian filtering technique is based on Bayes rule:

|  |  |
| --- | --- |
|  | (1) |

where  refers to prior probability, a hypothesis that is devoid of any information other than the model, a differential equation in state space form, that governs it.  refers to the probability of an observation, or the confidence of a sensor reading.  is the total probability of the sensor's state space.  normalizes the estimate such that . is the posterior probability based on the hypothesis; sensor reading incorporated. Posterior probability or belief is an estimate of a guess (prior probability) given some evidence (such as sensor reading).

The recursive Bayesian filtering technique incorporate the Bayesian rule shown in (1) where:

|  |  |
| --- | --- |
|  | (2) |

|  |  |
| --- | --- |
|  | (3) |

Multiplying (2) and (3) will results in the following:

|  |  |
| --- | --- |
|  | (4) |

taking note on the exponent term of (4) an expanding the terms gives,

|  |  |
| --- | --- |
|  | (5) |

Dividing through by the coefficient of *x2* gives,

|  |  |
| --- | --- |
|  | (6) |

the new Gaussian functions resulted in this multiplication has the generic Gaussian function,

|  |  |
| --- | --- |
|  | (7) |

Comparing equation (6) with the exponent terms of equation (7) gives

|  |  |
| --- | --- |
|  | (8) |

|  |  |
| --- | --- |
|  | (9) |

whereis the mean of position prior to measurement update, is the mean of the measurements, is update mean of the position,  is the standard deviation of position prior to measurement update,  is the standard deviation of the measurement, and  is the updated standard deviation of the position. Noted In Figure 1 is the narrow resultant Gaussian from the multiplication of and: the posterior estimate has a higher probability and a smaller variance than the two. It is counterintuitive but Equation 5 and Equation 6 show that posterior estimate has a smaller variance and means occurring between the mean of the sensor reading and the mean of the prior estimate. Put differently, by using Bayes’ rule, two stochastic process can produce information gain provided that both prior estimate (prediction) and measurement update (evidence) are unimodal Gaussian.

This step is repeated at each motion vector instate to the robot and, thus, recursively used the posterior estimate of the previous motion as the new prior estimate. Figure 2 shows the recursion of the old posterior as the new prior estimate. Kalman filter uses information of the immediate antecedent estimate for current posterior estimate, and ignore earlier posterior estimates based on Markov assumption.

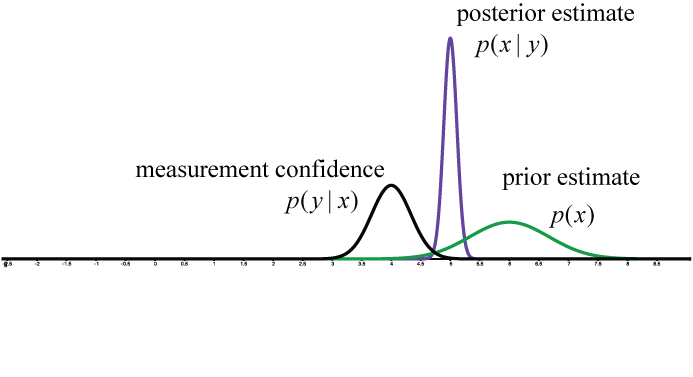


Figure 1 Posterior estimate after measurement update

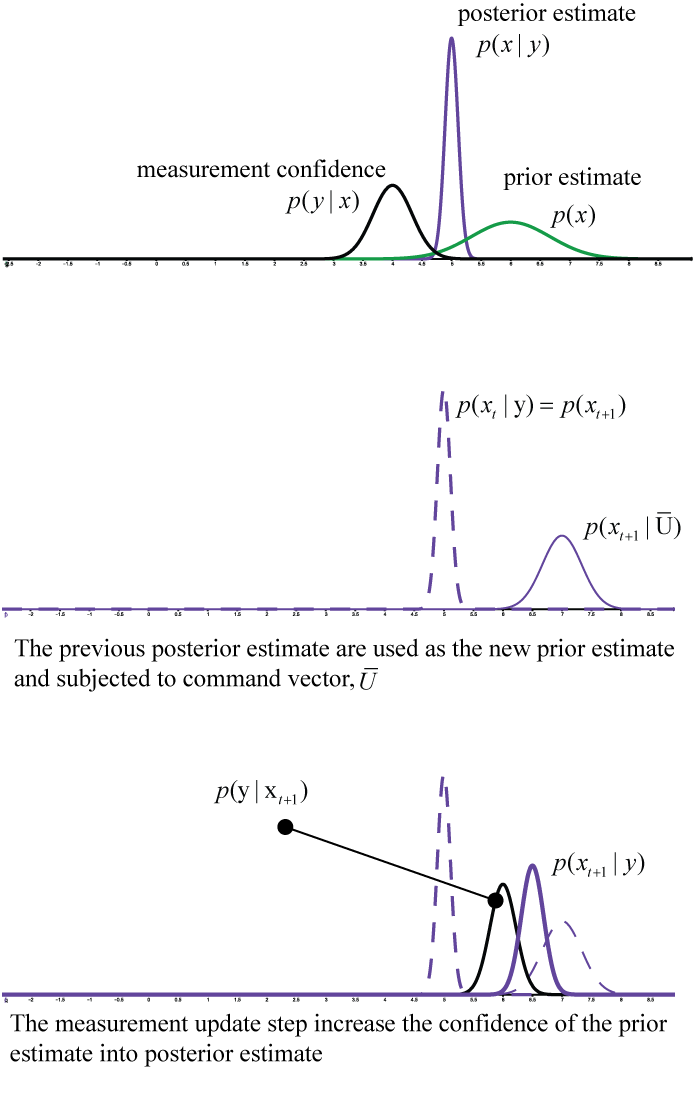


Figure 2 Recursive posterior estimation

Voxel contains information for three dimensional display, replicating three dimensional primitives by multiple instance of cubic boundaries [15]. For this research, voxel is regarded as a probabilistic entity that carries probabilistic parameter instead of just geometric representation. Each dimension of a voxel; width, height, and length, is encoded with variance of the current pose estimate output by the RBF. Figure 3 explains the structure of a voxel.

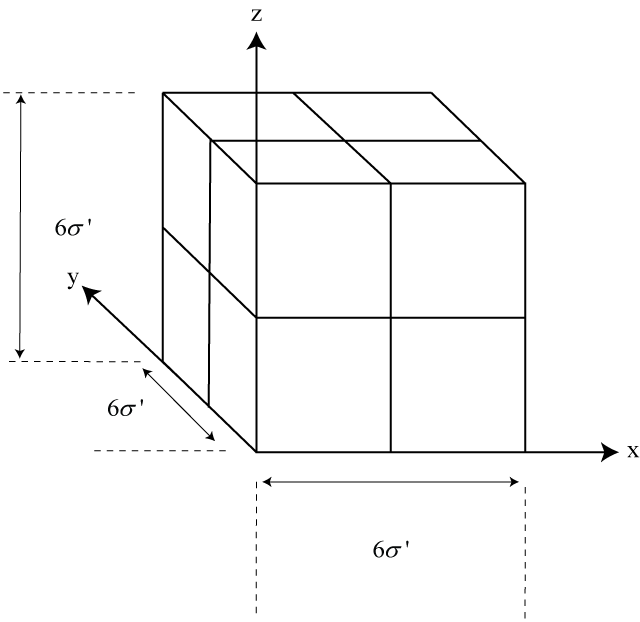


Figure 3 A single grid cell or voxel that has been voxelated based on the variance of the pose estimate

Each of the voxel is seeded with probability value given by a relaxed logit function:

|  |  |
| --- | --- |
|  | (10) |

where, , is the variance of the measurement, and, *n*, is the total point cloud incidence in the voxel based on a relaxed logit function where the classification of the voxel can be used (Hornung et al., 2013). Relaxed logit function in (10) is used because its characteristic to translate a discrete incidence into value between zero and unity.

Figure 4 explains the concept of embedding each voxel with probability value. Each point clouds encapsulate in the voxel is considered to have mean, *µ*, at the center of the voxel

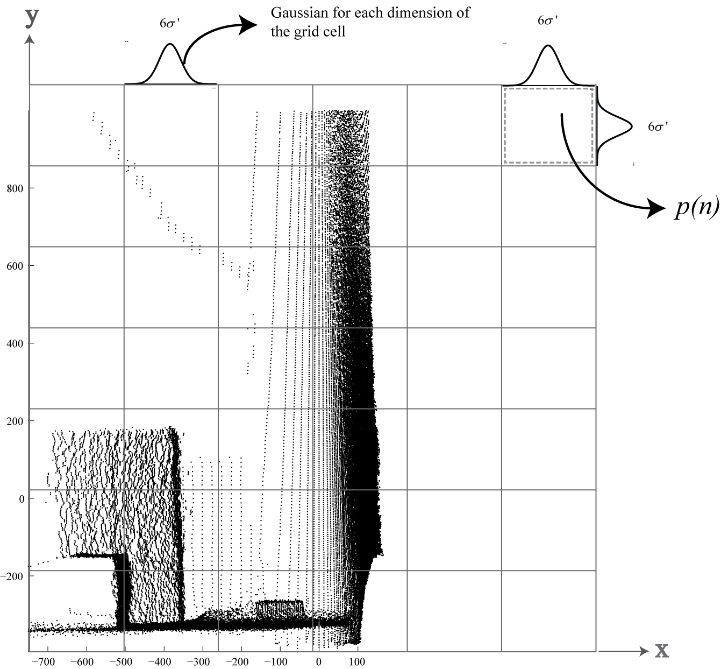


Figure 4 Grid map cell embedded with probabilistic parameter based on point clouds

To represent the model of the sensor and the voxelated grid map of the system, the position of the point cloud in space is described by Dirac delta function, which will give a sense of fixed frame for map building.

Given the position of point clouds in space, , the corresponding cell in the grid map is

|  |  |
| --- | --- |
|  | (11) |

where  is the voxel location, and  is the standard deviation of the pose estimation. Since the map is modelled on Gaussian normal distribution, each voxel has mean of zero, and thus to transform the location of the voxel, Dirac delta function is used,

|  |  |
| --- | --- |
|  | (12) |

where  is the Dirac delta function.

The position vector for each voxel is,

|  |  |
| --- | --- |
|  | (13) |

where, , is the position vector for, .

The grid map is represented by voxel, embedded with information of the variance of the system. Since each cell of the grid map is modelled based on the pose estimate of the robot by incorporating both the variance of the robot motion and the uncertainty of the sensor reading, the map changes every time the robot moves. Also, the location of the voxel is based on the transformation matrix obtained from Bayes’ rule filtering. Figure 5 explains the evolution of the voxel in the occupancy map. The depth cut-off of the grid map is not fixed and as the robot moves, it changes based on the predicted step of RBF.

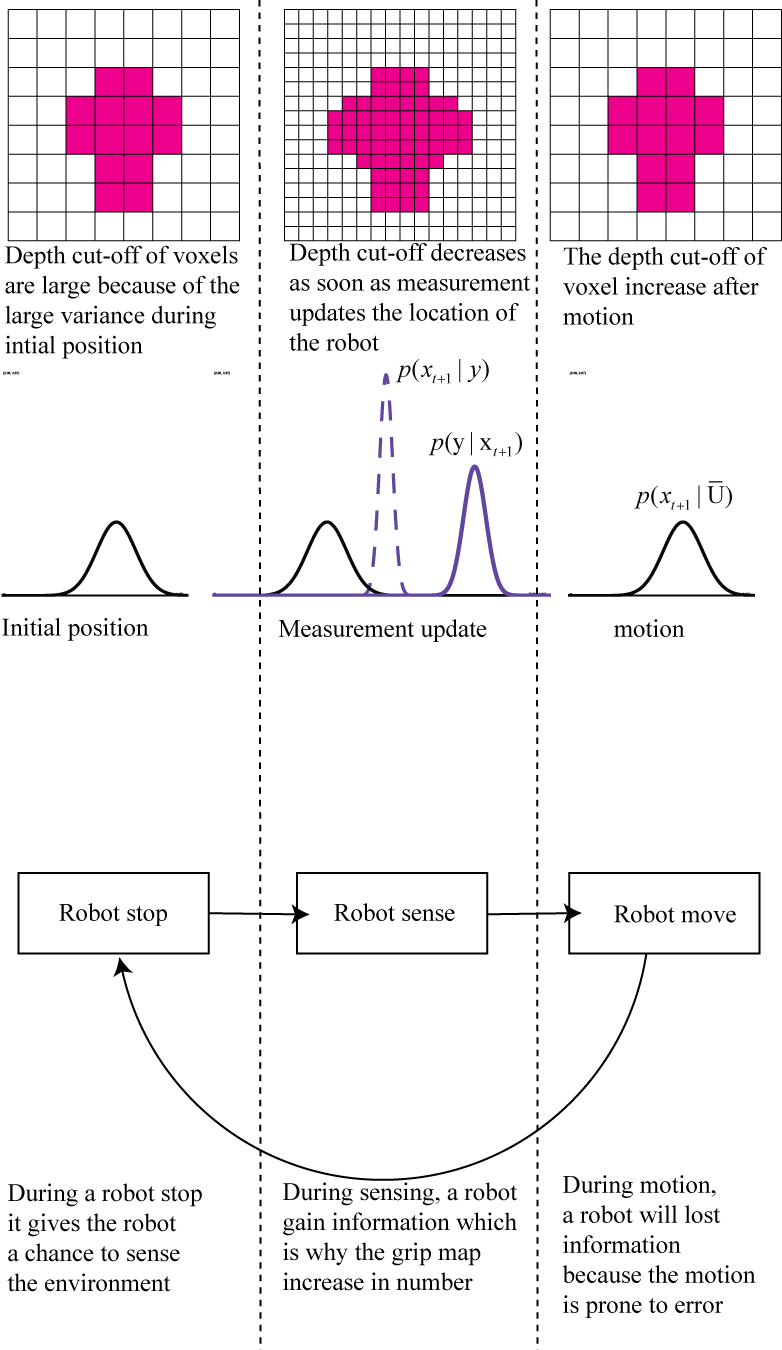


Figure 5 The information gain and information lost during measurement update and robot locomotion

EXPERIMENT PROTOCOL

The method of this experiment was to collect and register multiple scans, containing geometric information of a controlled environment, to produce a complete map. The experiment was decreed based on these guidelines:

1. The coordinate system complied with the right-hand rule as seen in Figure 6.

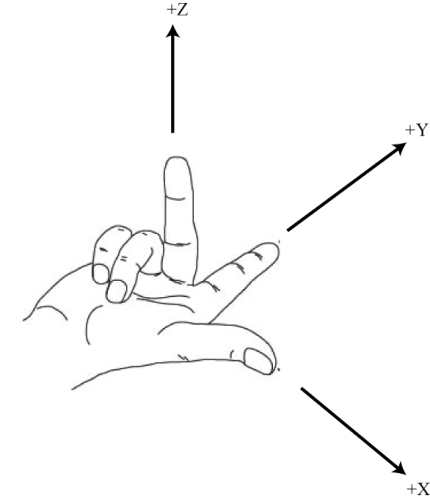


Figure 6 Coordinate system complied to right-hand rule

1. The controlled environment in this experiment is referred to by the prepared environment, explained in Figure 7, without the inclusion of other object out of the range of -750mm to 750mm in the *x*-axis, bigger than 2700mm in the *y*-axis, and -1000mm to 1000mm in the *z*-axis. Figure 8 depicts the set-up of the controlled environment. *Position three* and *Position four* shares the same point as are *Position five* and *Position two*. *Box A* and *Box B* are included to demonstrate occlusions in the environment.
2. The map registration was done in five different position and scan acquisition required the robot to stop moving and start scanning; the feature extraction algorithm was omitted and measurement of feature tracking was voxelized and measured by the experimenter
3. Each scans was subjected to robot motion at five unique planar positions. The simulated robot motion employed stop-move-stop locomotion to facilitate scanning. Figure 7 prescribes the location of each position in the controlled environment.
4. Each voxel was independent to the neighboring voxel.
5. The robot moved in a static enclosed environment; the motion was linear one dimensional locomotion.
6. Measurement variance for filtering was set to be 100 mm2 and the motion variance was at 2000 mm2.
7. The filtering technique for map registration used naïve Bayes’ rule filtering (BRF) which use Equation 7 without considering the old values of means and variance of the previous position and repeated with recursive Bayesian filtering technique (RBF).

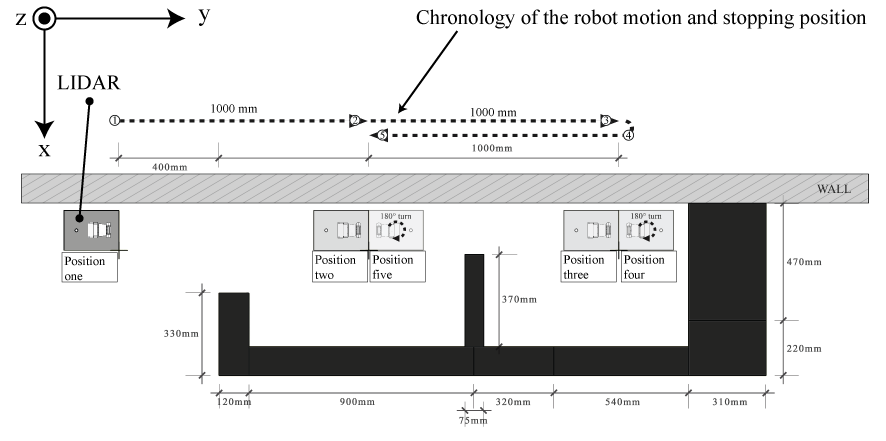


Figure 7 Plan view of the confined controlled environment

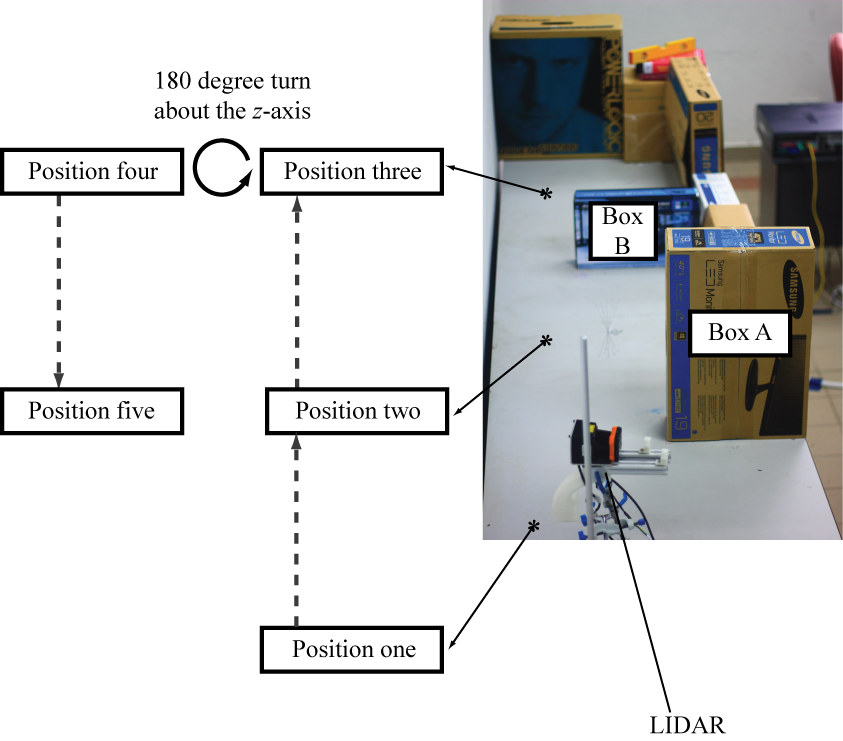


Figure 8 The set-up of the controlled environment

The environment in Figure 7 consists of boxes with various dimensions, stacked together, replicated an enclosed environment. Each boxes represent an obstacle and configured such that each scans collected from *Position one* to *Position five* introduce partial visualization of the environment. This would help in determining the success of the methods to register each scan. *Box A* was placed, for example, (Figure 9) in front of the laser in *Position one* such that the scans would consist of shadowed area. In Figure 9, the blind spot depicts an area that was not covered by the planar sweep and the yaw sweeping. A shadowed area is a discontinuity in metrical reading in point cloud image. Shadows results from the propagation of the infrared beam perpendicular to the plane of sweeping motion. Figure 10 illustrates the sweep motion of planar scanning and the propagation direction of the beam where, tinitial, occurs at the beginning of planar sweep scanning, t', occurs between the beginning of the sweep and the midpoint of the sweep, t'initial, occurs at the mid position of the sweep, and tfinal, occurs at the end of the sweep. A shadowed area is a limiting factor for complete environment mapping thus the robot must move to a different location. *Position four* introduce 180o turn to the LIDAR and would gain geometric information about the aforementioned shadowed area. The scanning concluded at position five where all the necessary data for complete map of this controlled environment were divided among five scans.

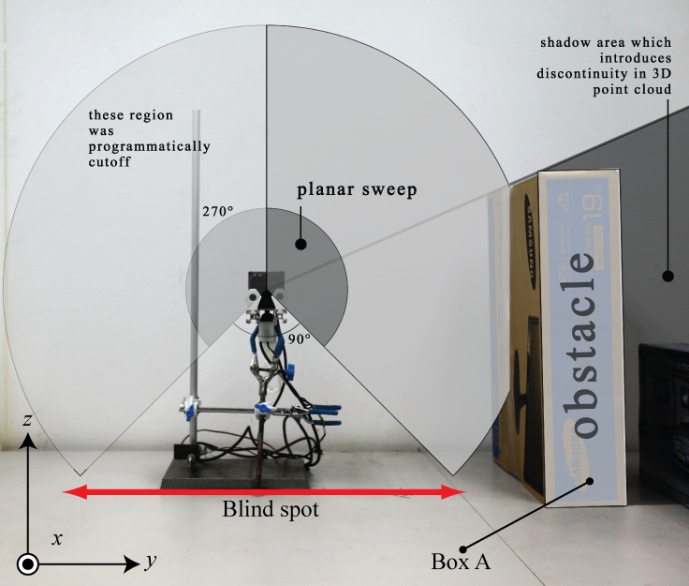


Figure 9 Shadowed area caused by obstacle, *Box A*

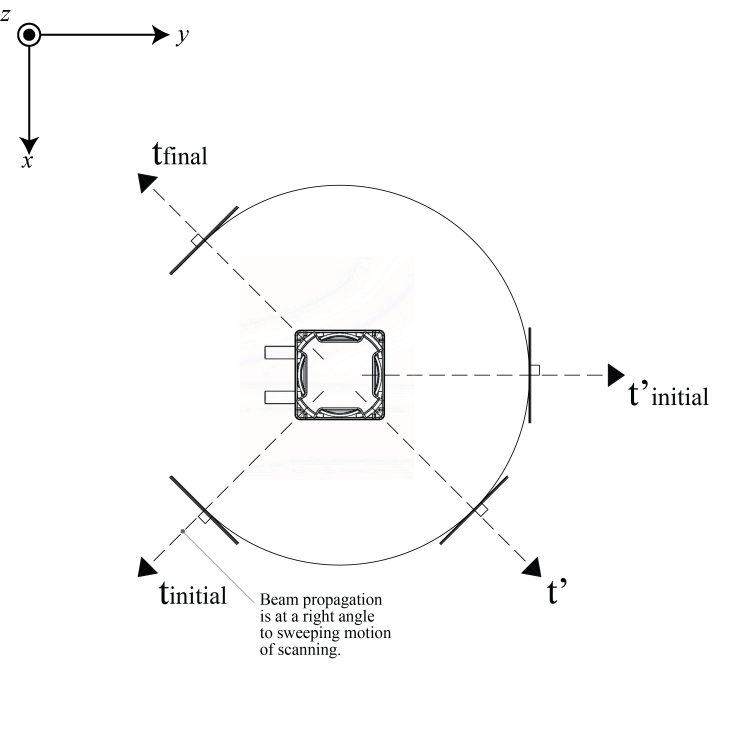


Figure 10 Beam propagation for LIDAR perpendicular to planar sweeping

The LIDAR was placed on a retort stand as prescribed in item (b) in the Introduction. The retort stand was placed to the designated position prescribed in the Figure 7, each followed by a scan acquisition. The value of the number on each position depicted the chronology of a pseudo-motion of a robot. Less care was taken in positioning the retort stand to the designated position simulating uncertainty in motion. This resonates with the real motion of a robot which is imprecise and prone to slippage. A scan was collected at each prescribed location.

RESULT

Figure 11 illustrates the position of the LIDAR in the controlled environment. Some features in the controlled environment were noted: as depicted in the figure, *Vertices A, B,* and *E* belong to *Box A,* and *Vertices C and F* belong to *Box B. Position five* is noted also in the figure with *Point D* addressing the position of the LIDAR. To reach *Position five*, the simulated robot was fed with a command vector to move 1000 mm forward at a heading of 180o. Vertices *E* and *F* were introduce in the figure to be used as a reference for comparing dimensional difference between maps.

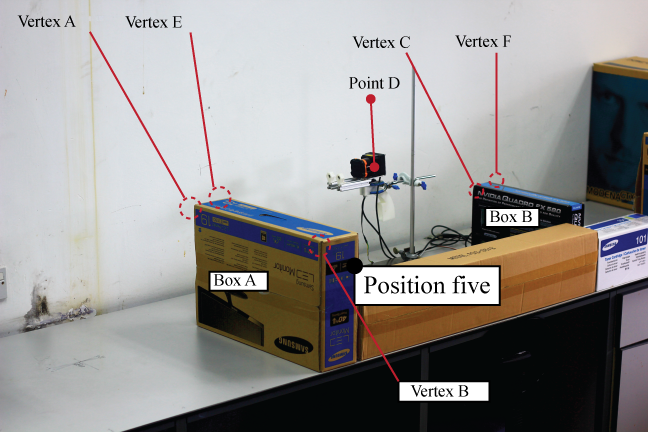


Figure 11 The scanner was placed at *Position five* concluding the scanning of the controlled environment

Figure 12 shows the probabilistic voxelated grid map produced through BRF. The isometric view showed huge misalignment between *Region two* and *Region three*. The misalignment was due to the naive approach of BRF where the update stage was not executed at each positional transition in the controlled environment. The filter used variance update at *Position one* throughout the exploration of the robot towards *Position five*. The approach was done to elucidate the concept of Bayes’ rule in trying to predict and estimate a posterior. Due to the naive approach of the BRF, dimensional information of *Box A* was also inaccurately reported.

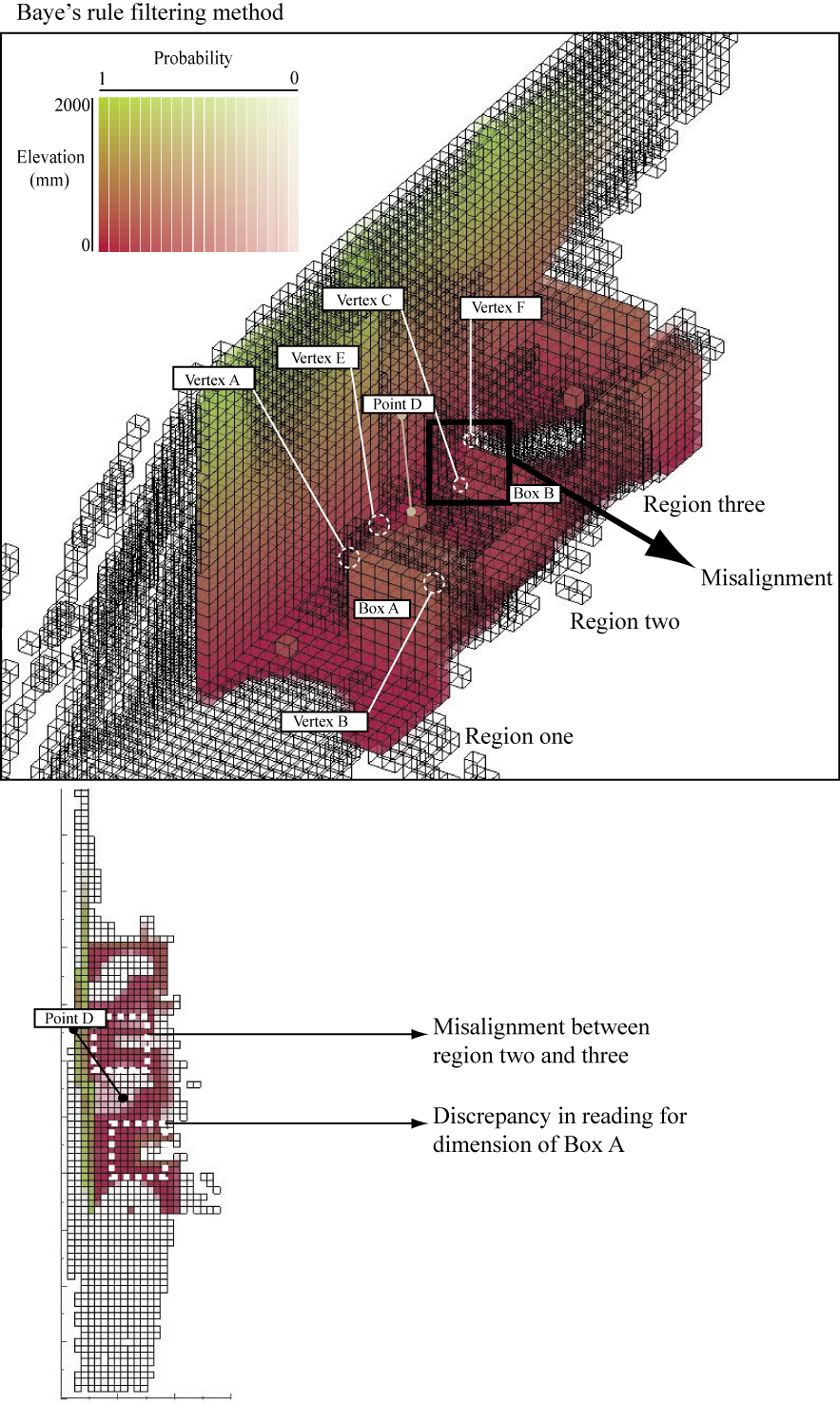


Figure 12 Probabilistic voxelated grid map (top) from using Bayes’ rule filtering and the top view of the map (bottom)

Figure 13 shows a consistent and more accurate representation of the controlled environment in voxelated grid map based on the localization method of RBF. *Region three* and *Region two* were aligned and the shadowed regions *S3, S2* and *S1* were populated by voxel cluster consist of high occupancy probability thus considered as an occupied space.

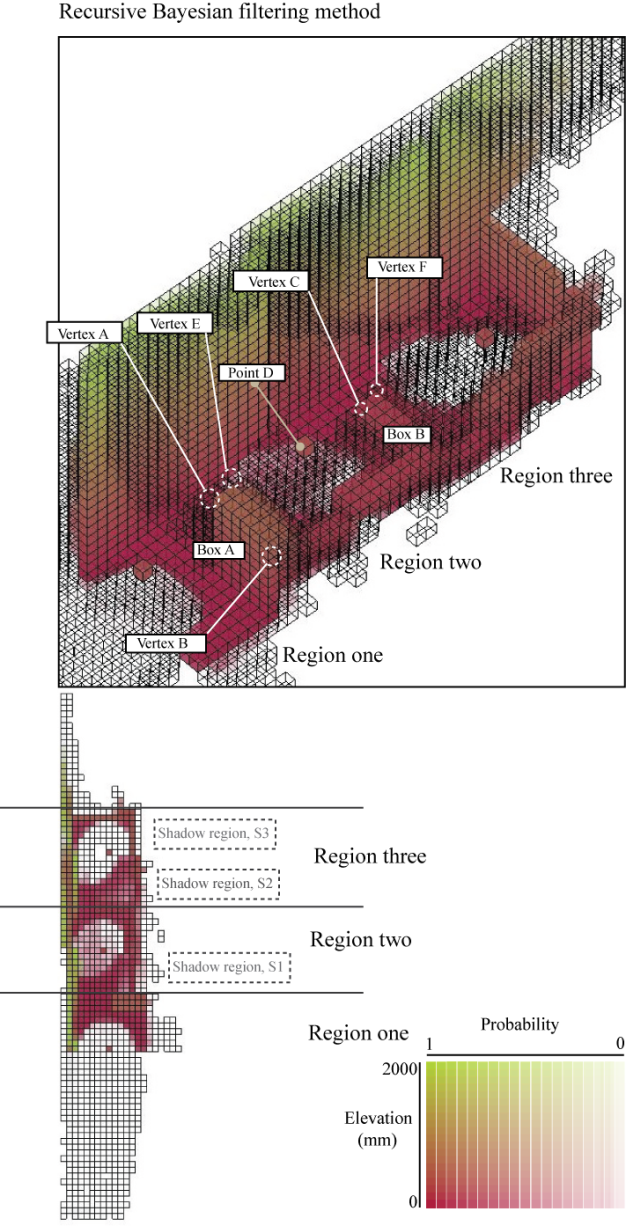


Figure 13 Probabilistic voxelated grid map at the final map registration (top) from using RBF and the top view (bottom) of the final grid map

Figure 14 shows the difference in geometric information portrayed in maps produce through BRF and RBF. The true dimension of edge, *AE*, of *Box A* was reported to be 120 mm. The discrepancy in this dimension was both noted in both maps. Under BRF, the map produce a dimension of 253 mm for edge *AE*, a 133 mm difference compared to the true dimension. Under RBF, the map produce a dimension of 116 mm, a 4 mm difference compared to the true dimension corresponding to 3.3 % difference.

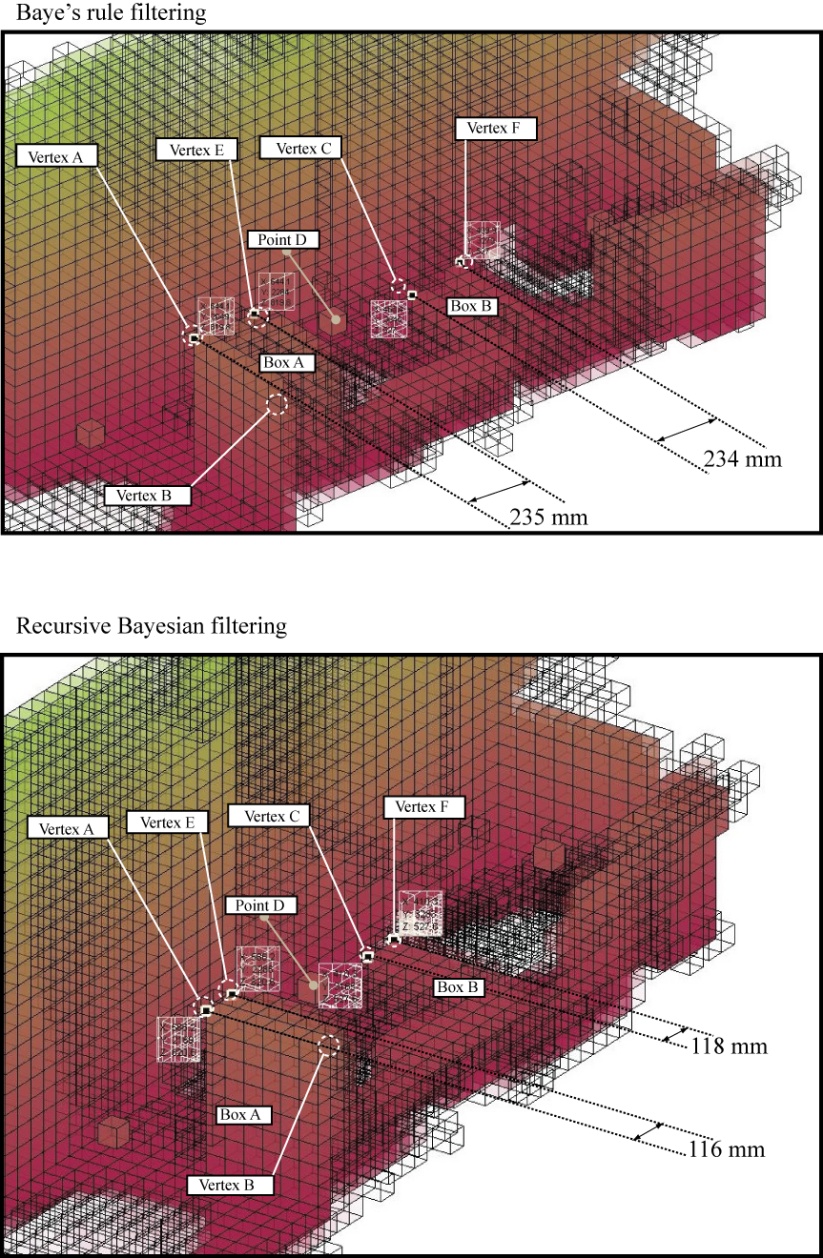


Figure 14 Dimension difference in edge *AE* and edge *CF* representing the dimensional depth of *Box A* and *Box B* respectively

The true dimension of edge *CF* was reported to be 75 mm. The dimension of edge, *CF*, observed in map produced under BRF was 234 mm; a large dimensional difference. The dimension of edge, *CF*, observed in map produced under RBF was 118 mm, a considerable large difference from the actual dimension. This error was due to the dimension of each voxel that could not cater the true value of *Box B*. At the end of the exploration done by the robot simulation, the measurement variance used to voxelate the scans was 95.4451 mm2. This would correspond to voxel depth of 58.61 mm. Since the true dimension of edge *CF* was 75 mm, the dimension could only be represented by two voxel depth of 118 mm.

This shows that the probabilistic voxelated map can perform well under recursive Bayesian filtering technique and the used of other variation of such filtering would give an esteem representation of the environment confined only by the measurement variance of the sensor. Table 1 summarizes the difference between the actual length and the length obtained from map produced through RBF technique. Figure 15 explains the configuration of all the edges *AB*, *AE*, *EG*, *GH*, *HC*, *CF*, *FI*, *IJ*, and *JK* introducing vertices *G*, *H*, *I*, *J*, and *K.* The average accuracy of these lengths obtained from RBF map was 0.87.

Table 1 Comparison of actual Length of multiple edges with the length obtained from map through RBF technique

|  |  |  |  |
| --- | --- | --- | --- |
| Edge | Actual length of the edge (mm) | Length of the edge from RBF map (mm) | Accuracy |
| AB | 330 | 351.7 | 0.93 |
| AE | 120 | 116 | 0.97 |
| EG | 215 | 175.8 | 0.82 |
| GH | 900 | 820 | 0.91 |
| HC | 370 | 352 | 0.95 |
| CF | 75 | 118 | 0.42 |
| FI | 370 | 410.6 | 0.89 |
| IJ | 785 | 820 | 0.96 |
| JK | 575 | 586.1 | 0.98 |

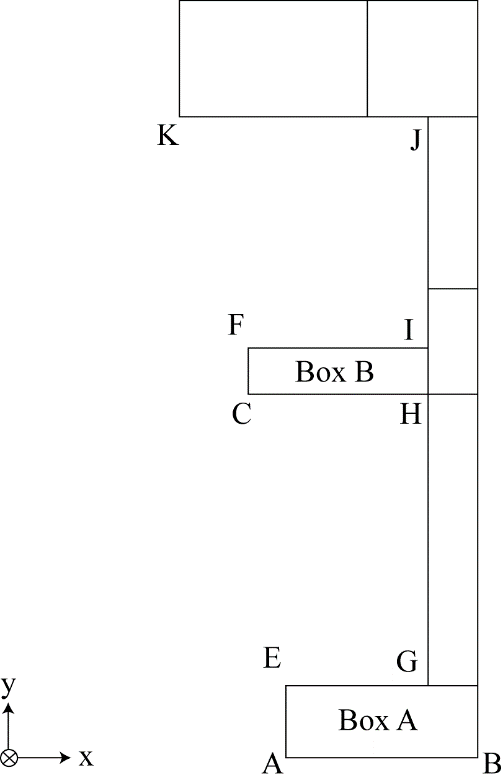


Figure 15 The top view of the controlled environment and the corresponding edges

Figure 16 shows the runtime of each voxelation at each position in the controlled environment. Slight increment in runtime is only due to the number of point clouds on each position. The increments are seen to converge at 0.23 s rather than increasing linearly. At *Position three*, the runtime decreased to 0.23 s.

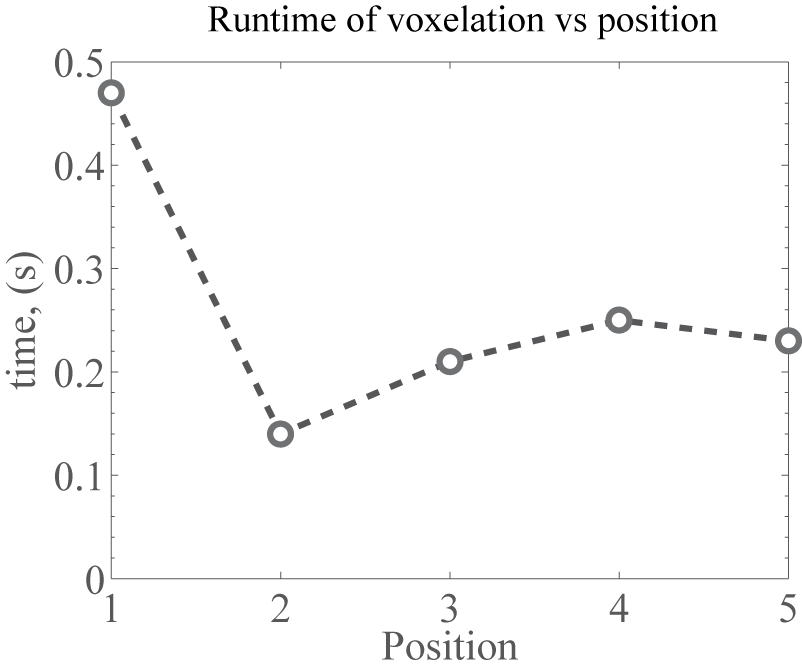


Figure 16 The runtime of voxelation for each position after scan registration

Figure 17 shows the increase in size for each transition in position. The size increase was expected due to the accrual nature of the map each time the position change. Each scan registered to the reference map, that has the global coordinate system, gained information and also caused increase in memory footprint. However, the compression of this information was seen in the dramatic decrease of file size from the raw point cloud to voxelated point cloud. From the figure, the file size between point clouds and voxelated point clouds in CSV format were seen to decrease by two decades in logarithmic scale. The size further decreased when the voxel was serialized to a binary format. This decrement in size could be explained by the nature of the voxelized grid cell where point clouds were aggregated using relax logit function. This aggregation of point clouds from the metrical data gave a conclusion of occupancy in space. This also showed that there was a decrease in information during the process of voxelizing a map, where points location were equated to be at the middle of the voxel. However, the frequency of occurrence of the point cloud data in each voxel was recorded. The occurrence frequency of point cloud in a voxel gave a probability representation of space since each point cloud represents the uncertainty of the sensor reading. The probabilistic voxelated format also reconfigured the information from point clouds, to a more logical representation of space. The representation of space in the form of probabilistic voxelated grid map would give a computational advantage to an autonomous robot because the information of the geometry of an environment would be easily queried due to the matrix structure of the grid map.

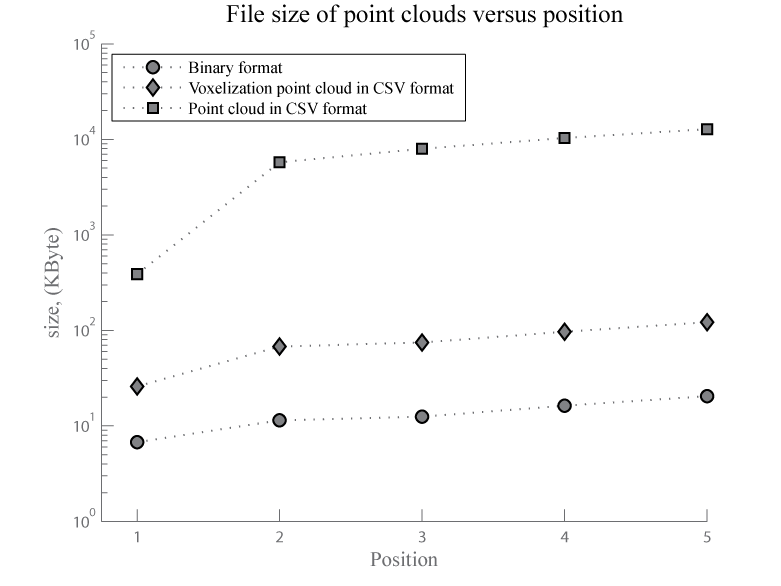


Figure 17 File size of three different serializations

CONCLUSION

In this research, an occupancy model called probabilistic voxelated grid map was developed to help redefine and extend the model of an unknown environment. The voxelization model was parameterized using the standard deviation of the Hokuyo UTM-30LX. The grid cell was modelled based on the voxelization of the sensor, incorporating relax logit function to translate the occurrence of occupancy into probability value. This method was tested by using one dimensional Kalman filter coined as recursive Bayesian filtering technique and compared against a non-recursive counterpart of the filter called Bayes’ rule filtering.

The result of this testing activities shows that probabilistic voxelated grid map represent the geometric information of the environment with 0.87 of average accuracy. This esteemed accuracy is enough to justify the method for use in autonomous robotics. Considering the probabilistic architecture of the model, the grid cell map in this research is compatible with most SLAM solution that uses Bayesian filters. Although research on grid cell map in three dimensional space has been pioneered, the advantage of probabilistic voxelated grid map lies in the use of continuous probability value to represent occupancy. With the accuracy of 0.87, the map can eliminate the need for other path planning method such as vector field path planning since each grid cell can act as a potential field in space.

The map was constructed by registering local maps into a global map. The runtime of this registration of map was observed to be below 0.3 s. The runtime of the map-building converges to 0.23 s which suggest the usability of this technique in real time system. Furthermore, compression of information of data from point clouds to voxelated grid cell map by two decades in logarithmic scale proved that the method introduce in this research were suited for autonomous agents. These results mirror the motivation of this research where the first objective was to develop probabilistic voxelated three dimensional grid map. To further exemplify this success, in Figure 5.15, the voxelated grid map managed to represent the geometric information of an unknown environment in three dimensional space without any misalignment. It was also in Section 5.8 that the second objective of this research was demonstrated. Converging runtime of 0.23 s and reduction of file footprint two to decades under Bayesian filtering techniques acclimate the efficiency of the method in representing unknown environment.

REFERENCES

[1] D. Borrmann, J. Elseberg, K. Lingemann, A. Nüchter, and J. Hertzberg, “Globally consistent 3D mapping with scan matching,” *Robot. Auton. Syst.*, vol. 56, no. 2, pp. 130–142, 2008.

[2] D. M. Cole and P. M. Newman, “Using laser range data for 3D SLAM in outdoor environments,” in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, 2006, pp. 1556–1563.

[3] N. Engelhard, F. Endres, J. Hess, J. Sturm, and W. Burgard, “Real-time 3D visual SLAM with a hand-held RGB-D camera,” in *Proc. of the RGB-D Workshop on 3D Perception in Robotics at the European Robotics Forum, Vasteras, Sweden*, 2011, vol. 2011.

[4] J. Weingarten and R. Siegwart, “EKF-based 3D SLAM for structured environment reconstruction,” in *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, 2005, pp. 3834–3839.

[5] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, “OctoMap: An efficient probabilistic 3D mapping framework based on octrees,” *Auton. Robots*, vol. 34, no. 3, pp. 189–206, 2013.

[6] E. B. Olson, “Real-time correlative scan matching,” in *IEEE International Conference on Robotics and Automation, 2009. ICRA ’09*, 2009, pp. 4387–4393.

[7] R. B. Rusu and S. Cousins, “3d is here: Point cloud library (pcl),” in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, 2011, pp. 1–4.

[8] G. Grisetti, R. Kummerle, C. Stachniss, U. Frese, and C. Hertzberg, “Hierarchical optimization on manifolds for online 2D and 3D mapping,” in *2010 IEEE International Conference on Robotics and Automation (ICRA)*, 2010, pp. 273–278.

[9] A. Elfes, “Using occupancy grids for mobile robot perception and navigation,” *Computer*, vol. 22, no. 6, pp. 46–57, Jun. 1989.

[10] H. P. Moravec, “Sensor fusion in certainty grids for mobile robots,” *AI Mag.*, vol. 9, no. 2, p. 61, 1988.

[11] R. Ray, V. Kumar, D. Banerji, and S. N. Shome, “Simultaneous Localisation and Image Intensity Based Occupancy Grid Map Building–A New Approach,” in *Intelligent Systems, Modelling and Simulation (ISMS), 2012 Third International Conference on*, 2012, pp. 143–148.

[12] A. Birk and S. Carpin, “Merging occupancy grid maps from multiple robots,” *Proc. IEEE*, vol. 94, no. 7, pp. 1384–1397, 2006.

[13] W. Burgard, D. Fox, H. Jans, C. Matenar, and S. Thrun, “Sonar-based mapping with mobile robots using EM,” in *MACHINE LEARNING-INTERNATIONAL WORKSHOP THEN CONFERENCE-*, 1999, pp. 67–76.

[14] K. M. Wurm, A. Hornung, M. Bennewitz, C. Stachniss, and W. Burgard, “OctoMap: A probabilistic, flexible, and compact 3D map representation for robotic systems,” in *Proc. of the ICRA 2010 workshop on best practice in 3D perception and modeling for mobile manipulation*, 2010, vol. 2.

[15] A. Kaufman, “Volume visualization,” *Vis. Comput.*, vol. 6, no. 1, pp. 1–1, 1990.

[16] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, “OctoMap: An efficient probabilistic 3D mapping framework based on octrees,” *Auton. Robots*, vol. 34, no. 3, pp. 189–206, 2013.