Improving Performance of 2D SLAM Methods by Complementing Kinect with Laser Scanner

K. Kamarudin, S.M. Mamduh, A.S.A Yeon, R. Visvanathan, A.Y.M Shakaff, A. Zakaria,
L.M. Kamarudin, N.A. Rahim
Centre of Excellence for Advanced Sensor Technology (CEASTech)
Universiti Malaysia Perlis, Arau, Perlis, Malaysia
Email: arul.unimap@gmail.com

***Abstract*— The feasibility of using Kinect sensor for 2D Simultaneous Localization and Mapping (SLAM) application has been widely studied. Researchers concluded that the acquired maps are often inaccurate due to the limited field of view of the sensor. Therefore in this work, we complemented the Kinect with a laser scanner and proposed a method to merge the data from both sensors. Two SLAM algorithms (i.e Gmapping and Hector SLAM) were tested using the method, in different environments. The results show that the method is able to detect multi-sized objects and produce more accurate map as compared to when using single sensor (i.e Kinect only or laser scanner only). Finally, the performance of the Gmapping and Hector SLAM are compared particularly in terms of the computational complexity and the map accuracy.**

Keywords—SLAM; Kinect; Laser Scanner; Robotics

I. INTRODUCTION

In an unexpected event requiring search and rescue operation, the environment's conditions are often unknown. The employment of robot to execute certain tasks in such scenario often requires the robot to predict its own position (i.e self-localization) and the environmental structure (i.e mapping). The use of GPS is usually not possible indoors due to poor coverage in a building and inaccurate measurement in small-scaled environment. Hence, the Simultaneous Localization and Mapping (SLAM) approach is preferred. SLAM operation can be described as the task of a robot in acquiring a map of its environment while simultaneously localizing itself relative to the map [1]. The problems have been claimed to be more complex than the "localization in a known map" and "mapping with known poses".

Numerous SLAM techniques have been developed by previous researchers utilizing different devices, including sonar sensors [2, 3], cameras [4-6] and laser scanners [7-10]; where, the laser scanner has been said to outperform its counterparts. The introduction of the Microsoft Kinect in 2010 and other 3D depth sensors allowed the possibility to integrate this type of sensor on the robot and perhaps able to replace or complement the widely used laser scanner [11]. The motivation is due to the extra dimensionality of the 3D depth sensor [12]. This feature allows the detection of objects of variable sizes and shapes; suggesting that a more reliable mapping and obstacle avoidance could be achieved. However, the narrow field of view of the Kinect's depth sensor has been reported by [12-14] as the limiting factor for producing a reliable SLAM results. The reason is that, the alignment between the limited scan endpoints and the recent map (i.e also called scan matching process) often failed to produce accurate pose estimation and map update.

This paper aims at extending our previous work in [12, 15] which analysed the feasibility of using Kinect for SLAM applications. We complemented the Kinect with a laser scanner and presents a method to merge the Kinect's 2D scan (obtained using [15]) and the laser scan. The merged scan was used as input for two SLAM techniques (i.e Gmapping and Hector SLAM) to perform mapping. The resulting maps were analysed and compared to the maps obtained using the Kinect alone or laser scanner alone; with the focus on the ability to detect objects of various sizes and the accuracy of map. In addition, the advantages and disadvantages of using each of the SLAM techniques are also discussed.

II. ROBOT AND SYSTEM

Fig. 1 shows the front view of the robot used in the research. The Kinect sensor and the laser scanner were mounted on top of the robot at its centre. Similar to the architecture presented in [12], the Robot Operating System (ROS) installed in virtual machine is used to run the SLAM algorithms. In addition, the method proposed in [15] is used to convert 3D depth sensor data into 2D obstacle locations and is denoted as Kinect Scan throughout the paper.

Fig. 1. The Robot equipped with Kinect, laser scanner, netbook and wireless router.

978-1-4673-7124-7/15/$31.00 ©2015 IEEE

III. METHOD TO MERGE KINECT'S 2D SCAN AND LASER SCAN

The open-source SLAM algorithms provided in Robot Operating System (ROS) allows only single input scan. Therefore the data from the Kinect and the laser scanner needs to be merged into a single scan. The Kinect with horizontal field of view (FOV) of 57° was mounted to face forward while the laser scanner with FOV of 270° was pointed backwards; giving a useful FOV of 327°.

Fig. 2 illustrates the method to merge both scans. First, the 2D Kinect's scan consisting of 640x1 arrays of X and Z coordinates are acquired using method in [15]. Then, the Cartesian-coordinates are converted into polar-coordinates, yielding the distance to the endpoint (i.e obstacle), r and the corresponding angle, θ. On the other hand, the laser scan has 769 scanning points with a resolution of 0.352°. These two data are combined so that the merged scan follows the laser's resolution of 0.352°. Since the Kinect's angle resolution is narrower, each of its scanning points are corrected to the nearest merged scan's interval. If multiple Kinect's scanning points fall onto the same direction, the one with closest endpoint is used since it is safer to assume the existence of nearer obstacle in that direction.

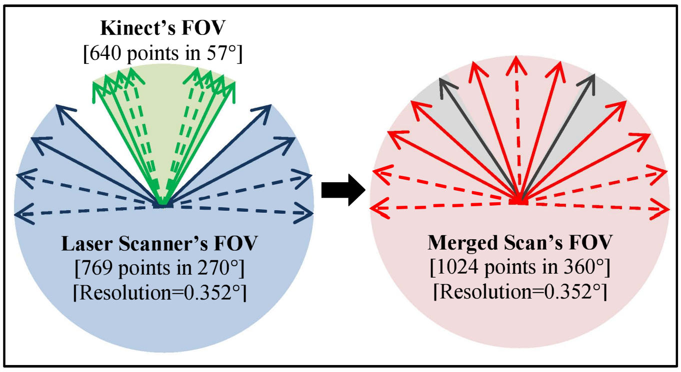

Fig. 2. The method proposed to merge Kinect scan and laser scan. The green, blue and red areas indicate the FOV of Kinect, Laser Scanner and Merged Scan respectively. The grey area is the blind spot area not covered by the sensors. Note that the illustration is not scaled to the real dimension.

The resulting scan therefore has 1024 points in 360° range but again with useful FOV of 327°. The scans which fall in the blind spot area (i.e 33° area not covered by Kinect and laser scanner) are represented by high value (i.e 10m); which will then be filtered by the SLAM algorithms. Note that the method proposed here requires the centers of the Kinect and the laser scanner to coincide in X and Z axis (i.e from top view); since the combination are done through polar coordinates.

IV. EXPERIMENTS

Several experiments have been conducted to study the performance of SLAM using our method (see Section III). Different types of environments have been tested and discussed in the following subsections. Note that the Kinect scan, laser scan and combined scan will be referred to as *kinect_scan*, *laser_scan* and *merged_scan* respectively.

A. Room With No Feature

First, we performed SLAM experiments in a (4.50 x 3.18)m room with no feature (i.e no object or furniture). The aim is to compare the accuracy of the map with respect to the actual dimension. While the robot moved in real-time, the odometry, *kinect_scan, laser_scan* and *merged_scan* were computed and logged (using ROSbag record). Then, the Gmapping and Hector SLAM were simulated (using ROSbag play) utilizing each of the scans. All the parameters for each SLAM technique were set to default values except the *sensor range.* This parameter was set to 0.6 to 6m for every scan type (i.e *kinect_scan, laser_scan* and *merged_scan*); following the useful range of Kinect depth sensor.

The robot was controlled remotely to move for 122 seconds; making an 8-shaped trajectory. This is to ensure the robot covers majority of the area as well as allowing the sensors to head to different directions along the way. Fig. 3 shows the comparison of the map obtained using Gmapping and Hector SLAM with different scan types. Note that the map may tilt according to robot's starting pose and due to the corrections made by the SLAM algorithm.

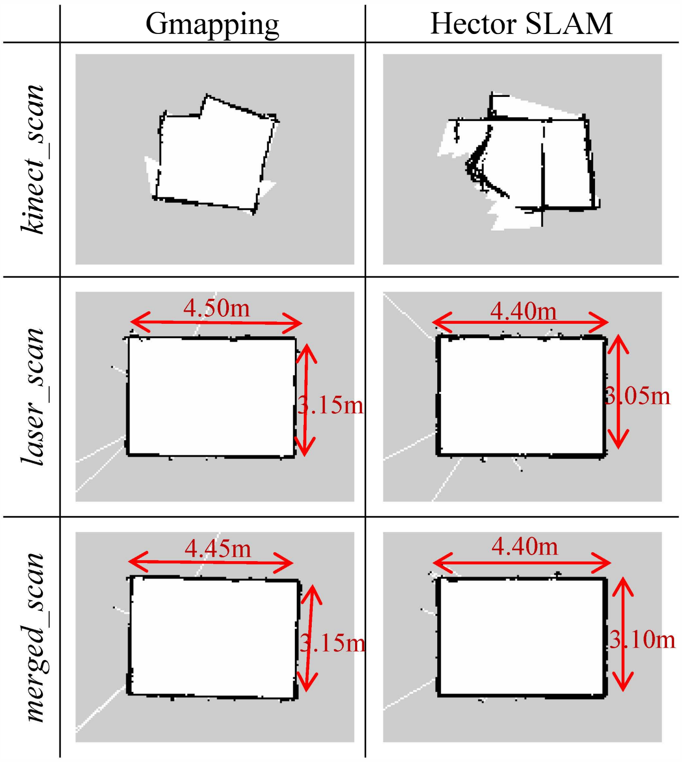

Fig. 3. Comparison of maps obtained using different SLAM techniques and scan types in [4.50m x 3.18m] room, without any features present. The red arrow and text indicate the dimension of the wall calculated between innermost black pixels and each pixel has a width of 0.05m.

The maps obtained using *kinect_scan* for both Gmapping and Hector SLAM are seen to deviate significantly from the actual dimension. This occurrence is due to the limited field of view (FOV) of Kinect that causes inaccurate scan matching computations (refer Section I).

On the other hand, the maps for *laser_scan* and *merged_scan* are highly accurate for both SLAM techniques; with the Gmapping using *laser_scan* gives the closest values to the actual dimension (i.e 4.50m x 3.18m). It could also be seen that the Hector SLAM produces slightly thicker wall as compared to Gmapping since it only relies on scan matching process. A few outliers in the *laser_scan* and *merged_scan* maps are also visible due to the noises in the laser scan during the SLAM operation.

Overall, the finding verifies that the method proposed to merge *kinect_scan* and *laser_scan* is able to provide reliable input for SLAM techniques and thus producing comparable map accuracy with respect to the widely used *laser_scan* alone.

B. Room With Known Features

Two additional trials were conducted at the same room but with additional features. The aim was to study the performance of SLAM in detecting objects of different size. A box of height 39cm (denoted as *taller_feature*) and another of height 19cm (denoted as *shorter_feature*) were placed in the room. The *shorter_feature* was below the detection range of the laser scanner while the *taller_feature* was within the sensor's FOV. The configuration of these two objects in each test is indicated in Fig. 4. The second test was performed to be about twice the period of the first one. The robot was remotely controlled to ensure that the trajectory covers around the features; avoiding blind spot areas.

Fig. 5 shows the comparison of the maps obtained using Gmapping and Hector SLAM with different scan types (i.e *kinect_scan*, *laser_scan* and *merged_scan*) in both experiments. It can be observed that the SLAM based on *kinect_scan* produced inaccurate map, similar to the result in the previous section. However, the *shorter_feature* is still visible in the map (see red arrows on the maps) indicating that the Kinect was able to detect the object.

The *laser_scan*'s maps for Test 1 are shown to be accurate for both Gmapping and Hector SLAM techniques. Conversely, in Test 2, the Gmapping produced slightly diverged map as compared to the Hector SLAM. The occurrence may be due to the fact that the Gmapping outputted bad particle (i.e map) after the twice longer operation in Test 2. More importantly, note that the *shorter_feature* is seen to be missing in all *laser_scan* maps; since the object was lower than the scanning height of the laser scanner. This weakness is not desired in SLAM as the robot may collide with the object or produce highly inaccurate result.

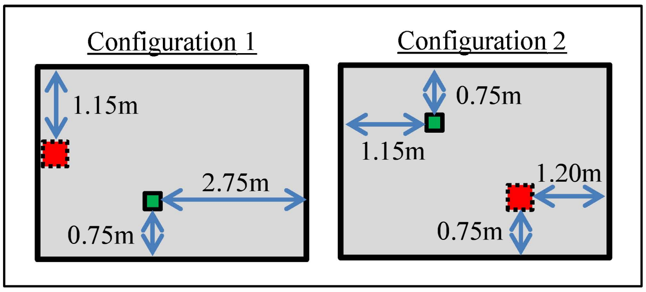

Fig. 4. Configuration of features in [4.50m x 3.18m] room in two different experiments. The green solid block indicates a box with size of $[l, w, h]$ = [0.27m,0.27m,0.39m] while the red dashed block represents [0.41m,0.41m,0.19m] box.

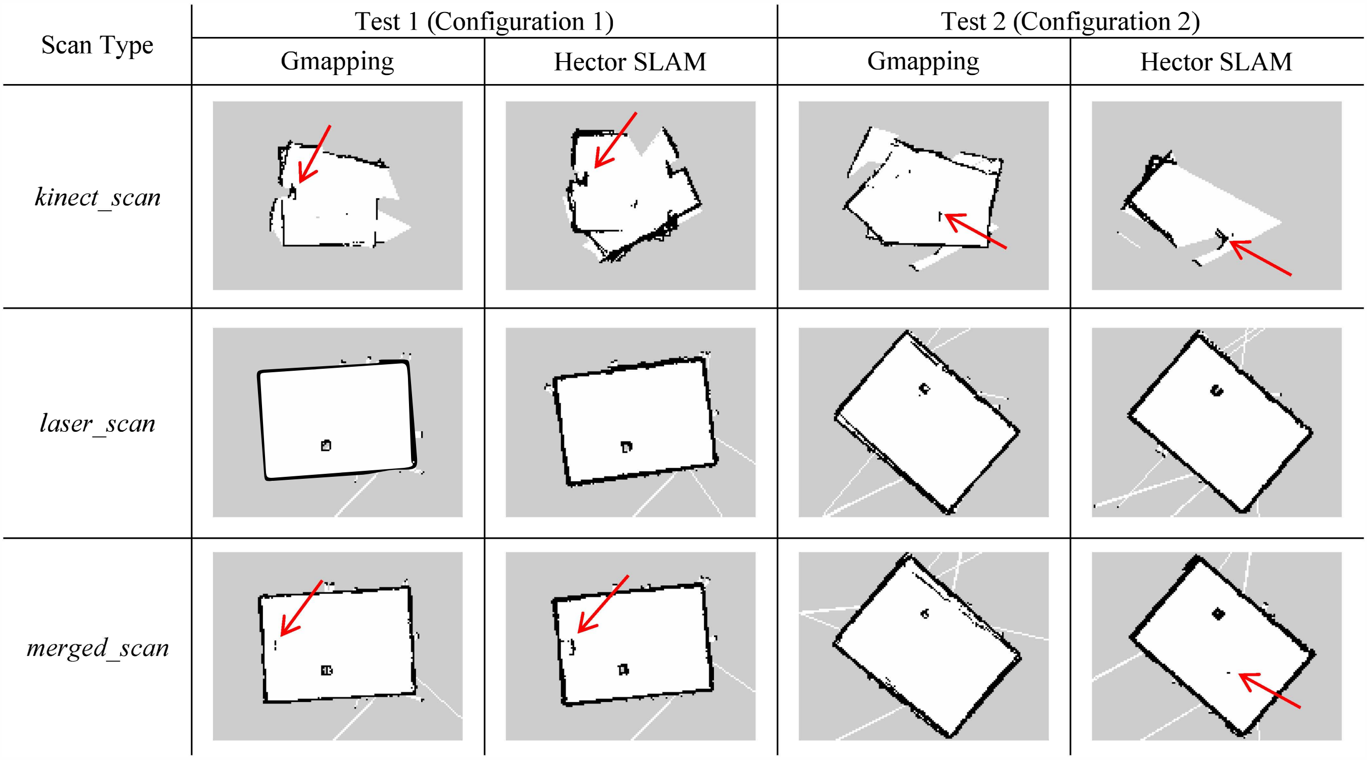

Fig. 5. Maps obtained using different SLAM tecniques, scan types and room's configuration. The red arrow indicates the *shorter_feature* being mapped.

The *merged_scan*'s maps in Test 1 are seen to be more accurate than the *laser_scan* maps; such that the *shorter_feature* is visible at the corresponding location. However, the result is less accurate in Test 2 where only a slight portion of *shorter_feature* is mapped using Hector SLAM and totally absent using Gmapping. This problem is due to the fact that the *shorter_feature* is only detectable in Kinect's FOV. When the feature falls into the laser scanner's FOV, it is not detected and thus assumed to be absent. The SLAM algorithms processed all these scenarios during runtime and the highest probability (either occupied or non-occupied) for each cell is selected. Since the *shorter_feature* in Test 1 is located near to the wall, the robot had been able to detect the feature more frequent than in Test 2; thus more portion of the feature is mapped.

Overall it can be concluded that the Hector SLAM using *merged_scan* produced the best map in term of accuracy and its ability to detect objects of different sizes. Although the *shorter_feature* is nearly missed in Test 2, the map can still indicate that an object exists at the corresponding location.

C. Corridor With Various Features

This section investigates the performance of SLAM techniques in real-world conditions, where various objects are present. Fig. 6 shows the partial drawing of the environment (i.e corridor) being tested. The grey blocks indicate the areas containing objects of various sizes and shapes.

The robot was controlled remotely to move around the corridor for 702 seconds; covering as many areas as possible. Similar to the previous experiments, the odometry, *kinect_scan*, *laser_scan* and *merged_scan* were logged during the robot's operation. Then the simulation of SLAM is performed based on the recorded data and using default parameters.

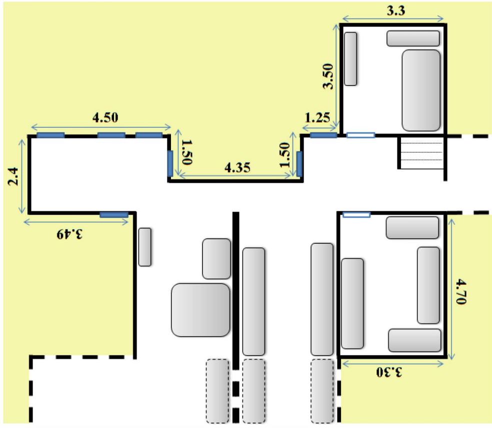

Fig. 6. Partial drawing of CEASTech's lab corridor (dimension in m) with grey blocks indicating the area containing objects of various sizes.

Fig. 7 shows the results obtained using Gmapping and Hector SLAM with different scan types (i.e *kinect_scan*, *laser_scan* and *merged_scan*). The maps computed based on *kinect_scan* are seen to be extremely inaccurate; again due to the limited FOV of the Kinect. The scattered and non-continuous map observed when using Hector SLAM is due to the failure of the algorithm to scan match; since it entirely depends on scan observation to predict pose.

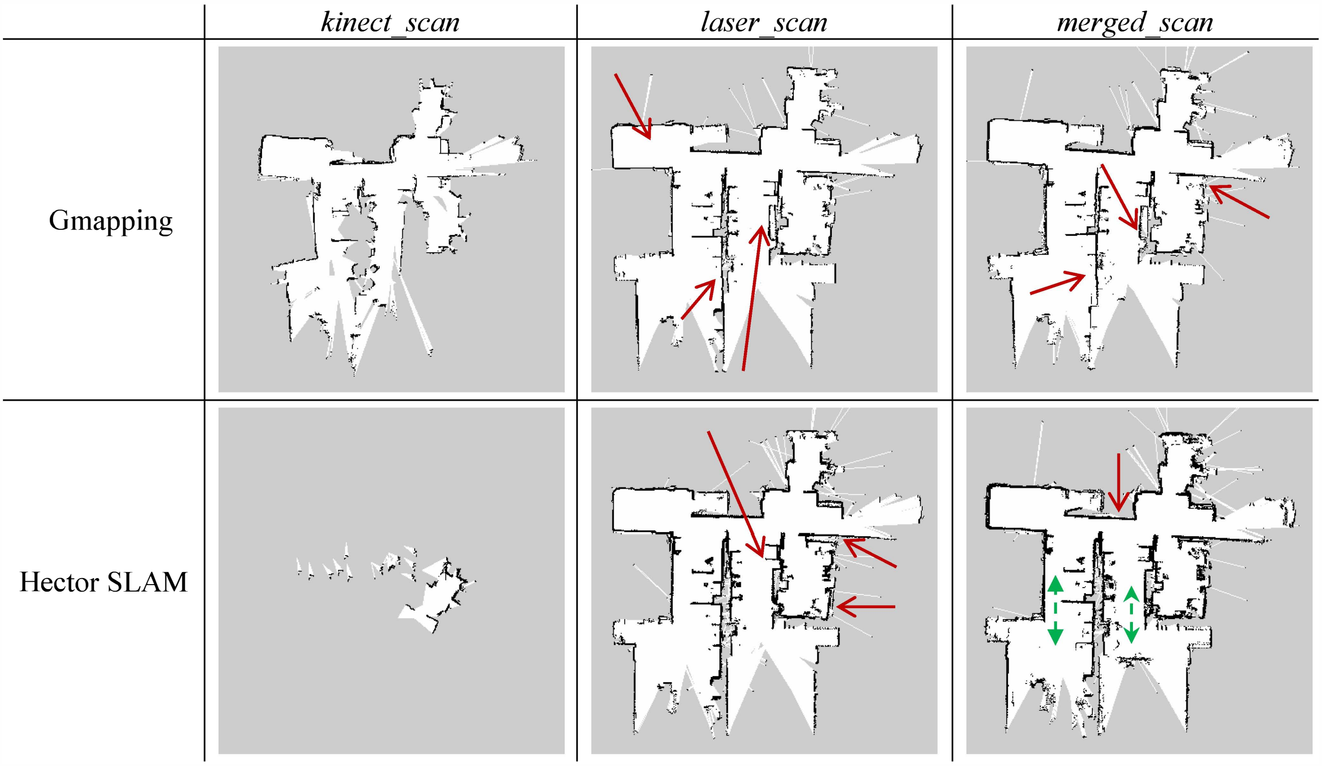

Fig. 7. Comparison of map obtained with different SLAM techniques and scan types in the corridor with various features. The red solid arrows indicate the skewed area or obstacle while green dashed arrows indicate the area of interest that contains multi-sized features.

The maps based on *laser_scan* are seen to be more accurate than the map based on *kinect_scan.* However, some of the area or walls are skewed with respect to the global map (indicated by the red arrows); with the Hector SLAM producing lesser number of this problem. Similar to the results in previous sections, the walls in the maps produced by Hector SLAM are thicker as compared to the Gmapping.

The *merged_scan's* maps are seen to indicate relatively more features than the *laser_scan's* maps for both the Gmapping and Hector SLAM; especially at the areas occupied by multi-sized objects (refer Fig. 6). This result verifies that the use of Kinect has been able to assist the robot in detecting short and variable shaped features; that were missed or inaccurately acquired by laser scanner. Interestingly, the merged_scan based Hector SLAM's map is shown to have the least skewed area and able to illustrate the highest number of features below the laser scanner detection range (especially at the areas indicated by green dashed arrows in Fig. 7).

V. Comparison Between Gmapping and Hector SLAM

To further study the performance of Gmapping and Hector SLAM, the total number of transform and map updates outputted by the algorithm (from Section C) were also recorded and shown in Table 1. The transform updates for Gmapping are found to be significantly less due to the fact that the Gmapping is based on complex particle filter algorithm that takes into account both the odometry and scan matching to compute the pose correction. In contrary, the Hector SLAM calculates the pose entirely based on scan matching alone; thus allowing an update rate close to the rate of the input scan (i.e *merged_scan*, *laser_scan* or *kinect_scan*).

It is important to note that the type of transform update outputted by the Gmapping and Hector SLAM are different. The Gmapping only outputs transformation (i.e correction) from the odometry frame to the map frame. Thus, to calculate the robot pose, the published transform needs to be combined with the odometry information. The main drawback is that, the estimated poses contain odometry error particularly during the interval between the transform updates. On the other hand, the Hector SLAM straightaway outputs the transformation from the map frame to the scanning frame (i.e robot frame in this case); hence could directly be considered as robot pose.

In addition, it can be seen that the total number of map updates for Gmapping are lower than Hector SLAM in all the experiments. The setting parameters for map update period for Gmapping and Hector SLAM were 5 seconds and 2 seconds respectively (i.e default values). Since the experiment was ran for 702 seconds, the number of map updates should be around 140 seconds for Gmapping and 351 for Hector SLAM. However, the Gmapping failed to achieve the targeted rate. The finding verifies that the algorithm is computationally more expensive and lacks the ability to update map at higher rate; which is an important factor in certain applications. Table 2 summarizes the comparison between Gmapping and Hector SLAM based on all the experiments conducted.

TABLE 1. Total number of transform and map update in each experiment

SLAM / Scan	Gmapping		Hector SLAM	
	Transform update	Map update	Transform update	Map update
kinect_scan	122	83	10419	353
laser_scan	122	83	10468	353
merged_scan	122	82	10467	353

TABLE 2. performances Comparison between Gmapping and Hector SLAM based on the experiments conducted.

SLAM / Factor	Gmapping	Hector SLAM
Mapping accuracy	Able to map the environment accurately when using *laser_scan* and *merged_scan*. Produced map with narrower wall but more skewed areas/features.	Able to accurately map the environment when using *laser_scan* or *merged_scan*. Produced map with thicker walls but less skewed areas/features.
Detection of various size and shaped features	Able to detect multi-sized features using *merged_scan*; but less accurate than Hector SLAM.	Better detection of multi-sized features using *merged_scan.*
Loop Closure	Able to close the loop since it is based on particle filter.	Unable to close the loop. During the operation, the acquired map can only be altered locally (i.e at the areas within FOV of latest scan).
Computational Complexity	Significantly higher due to the complex particle filter technique and the fact that it utilizes both scans and odometry in the computation. Unable to update the map at fast rate.	Significantly lower due to the simpler (Gauss-Newton) algorithm. Scan matching is used to estimate the pose and locally correct the map.
Pose Estimation	Outputs pose correction instead of real pose. The correction is updated every 5 to 6 seconds. Hence, during the interval, the erroneous odometry has to be used to estimate the pose.	Directly outputs the estimated pose. The update rate is significantly faster than Gmapping; close to the update rate of the input scan.

VI. CONCLUSIONS

Overall, the results of the experiments verify that the method proposed to combine the *kinect_scan* and *laser_scan* (i.e into *merged_scan*) have been able to provide reliable input to the SLAM algorithms. The most significant improvements with respect to *laser_scan* based SLAM are the ability to detect multi-sized obstacles and slightly improved map accuracy.

In general, the Gmapping and Hector SLAM have been able to produce accurate results in the experiments using the *laser_scan* and *merged_scan*; but failed when using the *kinect_scan*. This suggests that a wide and comprehensive FOV of the scan are the important factors to ensure reliable SLAM operation. While both SLAM algorithms have advantages and disadvantages, the Hector SLAM is found to outperform its counterpart in majority of the factors.

REFERENCES

[1] S. Thrun, W. Burgard, and D. Fox, Probabilistic robotics: MIT press, 2005.

[2] C. JinWoo, A. Sunghwan, and C. Wan Kyun, "Robust sonar feature detection for the SLAM of mobile robot," in Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on, Edmonton, Canada, 2005, pp. 3415-3420.

[3] A. Mallios, P. Ridao, D. Ribas, F. Maurelli, and Y. Petillot, "EKF-SLAM for AUV navigation under probabilistic sonar scan-matching," in Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on, Taipei, Taiwan, 2010, pp. 4404-4411.

[4] C. Mei, G. Sibley, M. Cummins, P. Newman, and I. Reid, "RSLAM: A System for Large-Scale Mapping in Constant-Time Using Stereo," International Journal of Computer Vision, vol. 94, pp. 198-214, 2011/09/01 2011.

[5] T. K. Marks, A. Howard, M. Bajracharya, G. W. Cottrell, and L. Matthies, "Gamma-SLAM: Using stereo vision and variance grid maps for SLAM in unstructured environments," in Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on, 2008, pp. 3717-3724.

[6] J. McDonald, M. Kaess, C. Cadena, J. Neira, and J. J. Leonard, "Real-time 6-DOF multi-session visual SLAM over large-scale environments," Robotics and Autonomous Systems, vol. 61, pp. 1144-1158, 10// 2013.

[7] S. Kohlbrecher, O. Von Stryk, J. Meyer, and U. Klingauf, "A flexible and scalable SLAM system with full 3D motion estimation," in Safety, Security, and Rescue Robotics (SSRR), 2011 IEEE International Symposium on, Kyoto, Japan, 2011, pp. 155-160.

[8] A. I. Eliazar and R. Parr, "DP-SLAM 2.0," in Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on, New Orleans, LA, USA, 2004, pp. 1314-1320 Vol.2.

[9] G. Grisetti, C. Stachniss, and W. Burgard, "Improved Techniques for Grid Mapping With Rao-Blackwellized Particle Filters," Robotics, IEEE Transactions on, vol. 23, pp. 34-46, 2007.

[10] J. Machado Santos, D. Portugal, and R. P. Rocha, "An evaluation of 2D SLAM techniques available in Robot Operating System," in Safety, Security, and Rescue Robotics (SSRR), 2013 IEEE International Symposium on, 2013, pp. 1-6.

[11] M. Viager, "Analysis of Kinect for mobile robots," Technical report, Technical University of Denmark, Lyngby, Denmark, Mar. 2011. http://www. scribd. com/doc/56470872/Analysis-of-Kinect-for-Mobile-Robots-unofficial-Kinect-data-sheet-on-page-272011.

[12] K. Kamarudin, S. M. Mamduh, A. Y. M. Shakaff, and A. Zakaria, "Performance analysis of the microsoft kinect sensor for 2D simultaneous localization and mapping (SLAM) techniques," Sensors (Switzerland), vol. 14, pp. 23365-23387, 2014.

[13] A. Oliver, S. Kang, B. C. W, and B. MacDonald, "Using the Kinect as a navigation sensor for mobile robotics," presented at the Proceedings of the 27th Conference on Image and Vision Computing New Zealand, Dunedin, New Zealand, 2012.

[14] S. Zug, F. Penzlin, A. Dietrich, N. Tran Tuan, and S. Albert, "Are laser scanners replaceable by Kinect sensors in robotic applications?," in Robotic and Sensors Environments (ROSE), 2012 IEEE International Symposium on, Magdeburg, Germany, 2012, pp. 144-149.

[15] K. Kamarudin, S. M. Mamduh, A. Y. M. Shakaff, S. M. Saad, A. Zakaria, A. H. Abdullah, et al., "Method to Convert Kinect's 3D Depth Data to a 2D Map for Indoor SLAM," in 9th IEEE Colloquium on Signal Processing and its Applications (CSPA 2013), Kuala Lumpur, 2013.