Method to convert Kinect's 3D depth data to a 2D map for indoor SLAM

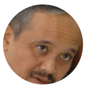
CONFERENCE PAPER · JANUARY 2013

DOI: 10.1109/CSPA.2013.6530050

7 AUTHORS, INCLUDING:

Kamarulzaman Kamarudin
Universiti Malaysia Perlis

Syed Muhammad Mamduh Syed Zakaria
Universiti Malaysia Perlis



ali yeon md shakaff
Universiti Malaysia Perlis

Ammar Zakaria
Universiti Malaysia Perlis

Method to Convert Kinect's 3D Depth Data to a 2D Map for Indoor SLAM

Kamarulzaman Kamarudin, Syed Muhammad Mamduh, Ali Yeon Md Shakaff, Shaharil Mad Saad, Ammar Zakaria, Abu Hassan Abdullah and Latifah Munirah Kamarudin
Centre of Excellence for Advanced Sensor Technology (CEASTech)
Universiti Malaysia Perlis, Arau, Perlis, Malaysia
Email: arul.unimap@gmail.com

***Abstract*— Mobile robotics has been strongly linked to localization and mapping especially for navigation purpose. A robot needs a sensor to see objects around it, avoid them and also map the surrounding area. The use of 1D and 2D proximity sensors such as ultrasonic sensor, sonar and laser range finder for area mapping is believed to be less effective since they do not provide information in Y or Z (horizontal and vertical) direction. The robot may miss an object due to its shape and position; thus increasing the risk of collision as well as inaccurate map. In this paper, a 3D visual device particularly Microsoft Kinect was used to perform area mapping. The 3D depth data from the device's depth sensor was retrieved and converted into 2D map using the presented method. A Graphical User Interface (GUI) was also implemented on the base station to depict the real-time map. It was found that the method applied has successfully mapped the potentially missing objects when using 1D or 2D sensor. The convincing results shown in this paper suggest that the Kinect is suitable for indoor SLAM application given that the device's limitations are solved.**

***Index Terms*—Robotics, Microsoft Kinect, Navigation, Indoor SLAM, Image Processing.**

I. Introduction

Navigation has been a subject of interest in mobile robotics over the past two decades. Even though a considerable amount of researches have been done on this particular application, various challenges and problems still remain. Researchers have been struggling to ensure the reliability of the navigation system since any failure may be harmful to human being. The navigation is a technique employed on a robot or a system to allow it to move from one location to another desired location [1]. This problem seems straightforward but in reality it is very complicated due to the non-ideal behavior of sensors and actuators. The term location can refer to the absolute location or at least relative position with respect to certain things.

The GPS has been greatly used for positioning method as it is able to directly provide the robot with its coordinate on earth. However, this technique is believed to be not practical for indoor since the error is high and the GPS satellite signal is weak inside the building. In order to solve this problem, a few researchers developed a new technique called indoor positioning; where static WIFI beacons were installed inside the building **[14][15]**. These beacons act the same way as the GPS's satellites by providing point to point distances for coordinate calculation. Despite good results obtained using this technique, it is believed to be impractical for exploring new areas due to the repetitive work required for beacons installment.

Another approach to achieve indoor navigation is Simultaneous Localization and Mapping (SLAM). The main idea of this technique is to leave the robot alone at an unknown location and let it move and build a consistent map of its surrounding **[2]**. There are numerous algorithms developed by past researchers such as Kalman Filter, Extended Kalman Filter, Fast SLAM and Occupancy Grid Map **[13]**. Most of the algorithms used odometry and proximity sensors to implement localization and mapping.

The use of 1D or 2D sensors for mapping is believed to be not efficient due to the possibility of missing an obstacle or even misinterpret its exact location. Figure 1 shows one of the scenarios where the robot can mistakenly determine the location of an object. The sensor will only see a part of the object rather than the whole shape. This phenomenon is undesirable in robotics application since it may cause a collision. A 3D sensor is believed be able to solve this limitation since the object's size and shape can be obtained. One of the most popular 3D sensors now is Microsoft Kinect which is commercially available at a price of less than USD150.

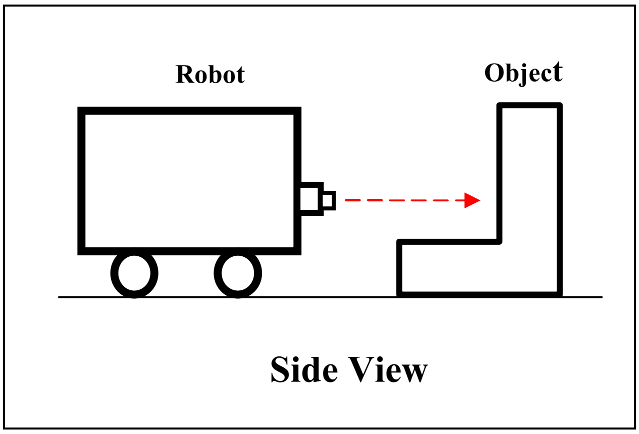


Fig. 1. Robot equipped with 1D or 2D sensor may mistakenly determine the location of the object.

This paper presents the feasibility of using Kinect with a mobile robot. A method is proposed and implemented to convert Kinect's depth data to a 2D area map. Finally, the results are evaluated to determine whether the Kinect is suitable for indoor SLAM application.

II. KINECT'S SPECIFICATIONS AND APPLICATIONS

Kinect was launched by Microsoft on November 2010 and it was initially designed for Microsoft Xbox 360 video game console. The Kinect functions as a motion-sensing input device which enables the game player to control and interact with the console through his movement. The device consists of several sensors particularly RGB Sensor, 3D Depth Sensor and Multi-array Microphones **[4][5]**.

The Kinect's depth sensor (which is the sensor of interest in this paper) produces depth image at 640x480 pixels (VGA size) **[4]**. The sensor has field of view of 58° horizontally and 45° vertically. The optimal operating range of the sensor is said to be between 0.8 to 3.5m although several researchers claims that it can be extended up to 0.6 to 6m. The Kinect's depth sensor is able to provide depth resolution of 1cm and spatial x/y resolution of 3mm at 2m distance from the sensor. The maximal stable transfer rate of the frame is up to 30 Hz depending on the driver or software used **[4][5]**.

Since the introduction of Kinect to the market, numerous drivers have been developed by different bodies to permit the Kinect sensors' interface. The Microsoft has released the first version of Windows SDK on 16 June 2011 allowing users to write application in C++, C# or .Net **[5]**. Another organization which is OpenNI developed open source drivers for Kinect to be used in multiple operating system including Windows, Linux and OSX **[8]**. The other famous group is OpenKinect which is an open community of over 2000 people who are interested in using Kinect with PCs and other devices. They jointly developed and maintain an open source software called Libfreenect for interfacing Kinect with Windows, Mac and Linux **[6]**.

Despite being used as a game controller, the functions of Kinect have been extended to variety of applications in robotics, healthcare, security and business. For example, El-laithy et al. performed experiments to study the limitations of using Kinect sensor on an autonomous ground vehicle. They claimed that the Kinect is not suitable for outdoor navigation during the day since intense sunlight can blind the Kinect's depth sensor [3]. Stowers et al. used the depth map from the Kinect sensor to control the altitude of quadrator helicopter. They claimed that the helicopter was successfully controlled in dynamic environment **[11].**

De Urturi Breton et al. developed a Kinect-based Windows application called KiMentia for helping individuals with Dementia problem. The application targeted to stimulate users' mental activities and at the same time improves their physical reaction **[12]**. Viager analyzed the use of Kinect for mobile robotics application. He came up with an unofficial data sheet for Kinect and a number of equations to calculate real 3D coordinate. Finally, he concluded that Kinect is a viable choice for mobile robot application **[9].**

III. SYSTEM OVERVIEW AND IMPLEMENTATION

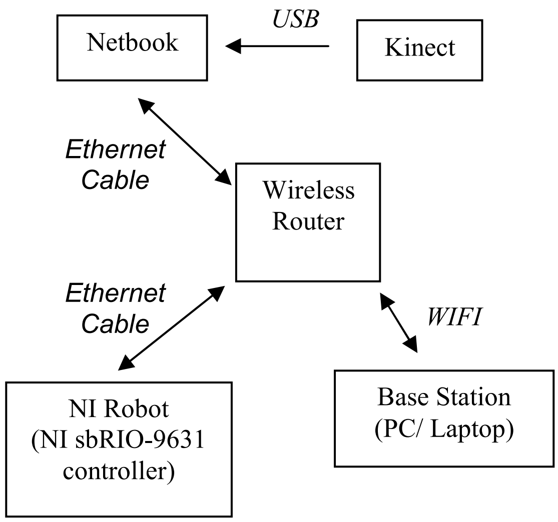


Fig. 2. Hardware Setup of the Whole System.

Figure 2 displays the hardware setup of the whole system. National Instruments' (NI) robot (refer Fig. 3) is used for maneuvering purpose. The robot is equipped with NI single board controller (i.e sbRIO-9631) and is programmed using LabVIEW environment. In order to allow a remote access, a wireless router is fixed on the robot and connected to the controller through an Ethernet cable. Currently, the robot's movement can be manually controlled from the base station through TCP/IP protocol.

A netbook was placed on top of the robot to acquire data from the Kinect. The netbook was chosen for this purpose since it is compact and light weight, has a powerful processor, equipped with long-last battery, and is pre-installed with Windows 7 operating system. This device is also connected to the router to allow TCP/IP data transfer to and from the base station. A forward facing Kinect is place on top of the netbook and connected through a USB cable.

The base station which is a PC or a Laptop is linked to the router through WIFI to allow a wireless remote access to the whole system. The base station is programmed using LabVIEW and performs communication with the robot's controller and netbook. The device provides Graphical User Interface (GUI) for system monitoring and control. It also shows the 2D map of the surrounding area and will be used for processing algorithms in future development.

(a) (b)

Fig. 3. (a) Front and (b) rear view of NI robot. The robot is equipped with Netbook, Router, and Kinect. The robot can be controlled remotely from the base station through TCP/IP protocol.

IV. METHOD TO CONVERT 3D DEPTH DATA TO A 2D MAP

In order to interface Kinect, the Libfreenect driver and LabVIEW's wrapper (written by Ryan Gordon) [10] were installed on the netbook. The LabVIEW was used for this system since it simplifies data analysis and processing. The arrays and images obtained can be viewed and monitored during real time; reducing the time required for debugging and analysis.

Figure 4 shows the RGB image obtained from the Kinect's RGB sensor. This image is displayed only for the purpose of showing the view of the Kinect Sensor on the robot and thus eases the task of comparing the depth sensor's image as well as the processed images.

Firstly, the 11-bit raw data (which is an array of 640x480 elements) was retrieved from the Kinect. Then, the data was processed and converted into real depth as seen in Fig. 5. Note that the darker the pixels, the nearer the object is. The white pixels represent invalid data either because they are out of the Kinect's valid range (i.e 0.6m to 4m), shadow, reflection or transparency problem. It can be seen that the image is reasonably smooth despite containing irregular dark pixels representing the floor and the cabinet's glass doors. The problem is believed to occur due to the reflectance of those surfaces. Also, note that the windows appeared to be white since the sensor's infrared beam can travel through them.

Next, the X and Y coordinate that correspond to each depth pixel (i.e Z coordinate) were calculated using

$$X_{i,j} = \left(j - \frac{w}{2}\right) \times \frac{320}{w} \times M \times Z_{i,j} \tag{1}$$

$$Y_{i,j} = \left(i - \frac{h}{2}\right) \times \frac{240}{h} \times M \times Z_{i,j} \tag{2}$$

where i and j are the pixel's row and column number in the Z-array respectively, w and h are the width and the height of the Z-array and M is the NUI Camera Depth Image to Skeleton Multiplier Constant. These equations were obtained from the Microsoft SDK library for Kinect.

Fig. 4. Image obtained from the Kinect's RGB Camera. This image is displayed only for the purpose of showing the view of Kinect Sensor on the robot.

Fig. 5. Image obtained by converting the depth array to grayscale.

Now, there are three arrays of 640x480 pixels representing the real X-Y-Z coordinate. In order to solve the floor problem stated previously, the pixels located below 26cm were replaced with infinity value indicating that the data is invalid. This value was chosen since the Kinect's depth sensor was positioned 28cm above the floor and the 2cm difference was added to avoid noises. We also eliminated the pixels above the height of the robot as they are not considered as obstacles. As a result, a smooth image (refer Fig. 6) is obtained in which the dark pixels considered as obstacles to the robot. Again, the darker the pixels, the nearer the obstacles are.

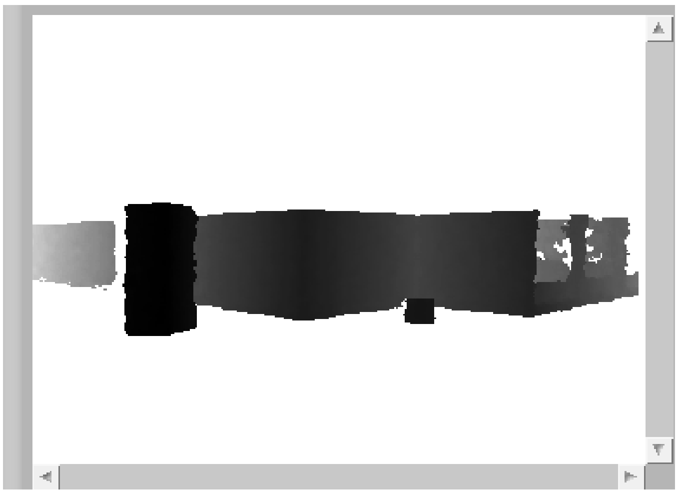


Fig. 6. Image obtained after eliminating the unnecessary pixels. The remaining dark pixels are considered as 3D obstacles to the robot.

Next the minimum element in each column of the Z-array was selected such that

$$Z'_j = min\ (Z_{0,j}, Z_{1,j} \dots Z_{479,j}) \qquad (3)$$

where j is the respective column number. The X locations for the corresponding Z-elements were also selected giving another 640x1 array. These two arrays indicate the Z and X coordinates of the nearest pixels and can also be regarded as 2D obstacle locations as obtained when using a laser scanner. The minimum-selection method in Eq. 3 was implemented to avoid false detection of obstacle and minimize processing power and time. Also note that the Y-Coordinate was ignored as the robot can only move in X and Z direction. The arrays have already represented the nearest obstacle locations no matter where their vertical positions are.

The 2D obstacle locations were then transferred to the base station through TCP/IP protocol. We managed to get a transfer rate of less than 1ms since there were only 1280 16-bit integers being transmitted through the 300Mbps wireless router. The base station then processed the data and drew the robot and the obstacles as shown in Fig. 7. Note that the locations of the obstacles are relative to the Kinect's point of view.

It can be seen that the small cylinder is successfully detected by the robot. This object may not be discovered if the 1D or 2D device was used. Also note that the wall behind the box and the cylinder is not visible since only the nearest obstacle was mapped (refer Eq. 3). The backside obstacles were ignored because an unseen object may be present between the box and the wall. In addition, observe that the further the object, the further the points are separated. This is due to the fact that the resolutions of X-Y-Z coordinate increases as the distance increases.

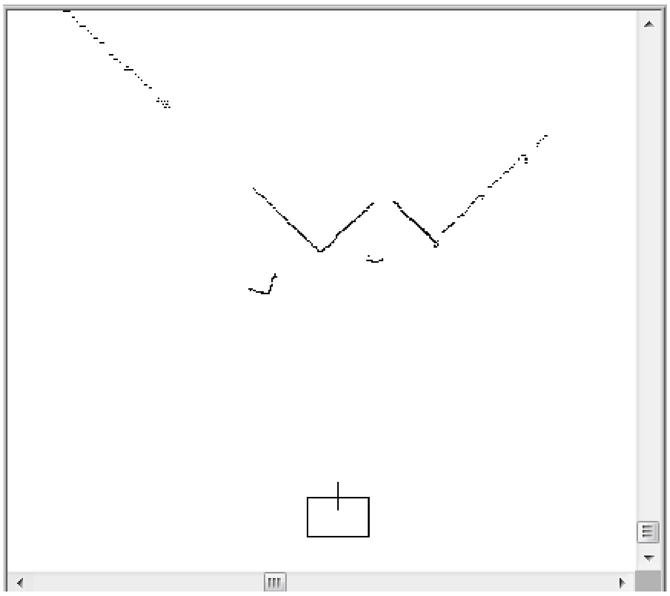


Fig. 7. 2D map as depicted on the base station.

V. CONCLUSION AND FUTURE WORK

In this paper, we proposed a method to convert the Kinect's 3D depth data to a 2D area map. The results shown prove that the conversion method has been able to map potentially missing object when using 1D or 2D sensors. The map obtained also was smooth and consistent suggesting that the proposed method is suitable for SLAM algorithms which utilize 2D obstacle locations.

However, the Kinect's problem in detecting the transparent and high reflection surfaces is still to be solved. Further processing methods needs to be developed so that the robot is capable of identifying these types of surfaces. A temporary solution is to install an additional ultrasonic sensor on the robot since it uses sound navigation and thus will be able to detect the surfaces. This device can also help in providing short-ranged vision to the robot.

Extensive tests and experiments are yet to be done to ensure that the Kinect and the proposed method are reliable under different conditions such as during robot's movement, measuring glass-typed building and moving around on uneven surface area. Another important issue to be considered is the huge memory requirement of SLAM application. Among the proposed solutions are to reduce the size of the X-Y-Z arrays and decrease the resolution of the map. These solutions will also improve the repetitive data transfer and processing time.

ACKNOWLEDGMENT

The open-source Kinect libraries were provided by the OpenKinect community and the LabVIEW wrapper was made available by Ryan Gordon. The project's equipment and facilities were provided by University Malaysia Perlis (UniMAP). The authors also thank to everybody who involved in the implementation and development of the project. Kamarulzaman acknowledges the financial sponsorship under the Academic Staff Training Scheme provided by UniMAP and Ministry of Higher Education, Malaysia (MOHE).

REFERENCES

[1] Siegwart, Roland, and Illah R. Nourbakhsh. *Introduction to autonomous mobile robots*. MIT press, 2004.

[2] Schindhelm, C.K. "Evaluating SLAM Approaches for Microsoft Kinect." In *ICWMC 2012, The Eighth International Conference on Wireless and Mobile Communications*, pp. 402-407. 2012.

[3] El-laithy, R.A.; Jidong Huang; Yeh, M.; , "Study on the use of Microsoft Kinect for robotics applications," *Position Location and Navigation Symposium (PLANS), 2012 IEEE/ION* , vol., no., pp.1280-1288, 23-26 April 2012

[4] PrimeSense, "The PrimeSense 3D Awareness Sensor" datasheet, 2012

[5] Microsoft. "Kinect for Windows." Internet: http://www.microsoft.com/en-us/kinectforwindows, 2012 [Jan. 10, 2013].

[6] Openkinect. "Welcome to the OpenKinect project." Internet: http://openkinect.org/wiki/Main_Page/, Mar. 7, 2012 [Jan. 10, 2013].

[7] ROS.org. "Kinect Calibration/Technical." Internet: http://www.ros.org/wiki/kinect_calibration/technical/, Dec. 27 2012 [Jan. 10, 2013].

[8] OpenNI. "The standard framework for 3D sensing." Internet: http://www.openni.org/, 2013 [Jan. 10, 2013].

[9] Viager, M. "Analysis of Kinect for Mobile Robots." Technical report, Technical University of Denmark, Lyngby, Denmark, Mar. 2011

[10] Ryan Gordon. "Kinect for LabView – Part 2." Internet: http://ryangordon.net/2011/01/20/kinect-for-labview-part-2/, Jan. 20, 2011. [Jan. 10, 2013].

[11] Stowers, J.; Hayes, M.; Bainbridge-Smith, A.; , "Altitude control of a quadrotor helicopter using depth map from Microsoft Kinect sensor," *Mechatronics (ICM), 2011 IEEE International Conference on* , vol., no., pp.358-362, 13-15 April 2011

[12] de Urturi Breton, Z.S.; Zapirain, B.G.; Zorrilla, A.M.; , "KiMentia: Kinect based tool to help cognitive stimulation for individuals with dementia," *e-Health Networking, Applications and Services (Healthcom), 2012 IEEE 14th International Conference on* , vol., no., pp.325-328, 10-13 Oct. 2012

[13] Hiebert-Treuer, Bradley. "An Introduction to Robot SLAM (Simultaneous Localization And Mapping)." *Bachelor of Arts in Computer Science-Middlebury College. Middlebury, Vermount, USA*, 2007.

[14] Ning Chang; Rashidzadeh, R.; Ahmadi, M.; , "Robust indoor positioning using differential wi-fi access points," *Consumer Electronics, IEEE Transactions on* , vol.56, no.3, pp.1860-1867, Aug. 2010

[15] Mazuelas, S.; Bahillo, A.; Lorenzo, R.M.; Fernandez, P.; Lago, F.A.; Garcia, E.; Blas, J.; Abril, E.J.; , "Robust Indoor Positioning Provided by Real-Time RSSI Values in Unmodified WLAN Networks," *Selected Topics in Signal Processing, IEEE Journal of* , vol.3, no.5, pp.821-831, Oct. 2009