Automatic Docking with Obstacle Avoidance of a Differential Wheel Mobile Robot

Warin Poomarin, Ratchatin Chancharoen, and Viboon Sangveraphunsiri
Chulalongkorn University, Bangkok, Thailand
Email: warren.rice.ca@gmail.com, {Ratchatin.C, Viboon.S}@ chula.ac.th

***Abstract*—The paper presents a technique and a system that can be used to smoothly dock a differential drive mobile robot. The system consists of only a robot, a RGBD camera, and a notebook computer which are connected via USB cable. The RGBD camera with the proposed processing technique is capable of determining the Cartesian positions of a home base, obstacles, a floor, and a wall. They are then plotted on a 2D grip map where a free trajectory towards the home based can be constructed. This trajectory is then used to guide the robot to dock at the home base. Experimental result is used to demonstrate the applicable of the proposed technique.**

***Index Terms*—automatic docking, a mobile robot, navigation and guidance**

I. Introduction

Autonomous mobile robots are now widely used as service robots in offices [1] and houses [2] as their intelligence are improved and their costs are significantly reduced. An autonomous vacuum robot, such as iRobot's Roomba, is an example of a service robot that works at our living space while surveillance [3] or telepresence [4] robots are also in service at office or hospital. As autonomous robots, they should be capable of self-maintenance and indoor navigation [5]. The paper presents our work to develop an automatic docking technique of a mobile robot where obstacles are presented using commercially available devices. These features allow an autonomous robot to go back to its base for battery charging and routine maintenance. In this work, only the commercially available devices are used including iRobot's Create and Xbox's Kinect camera which are connected to a computer notebook via only USB cables. The proposed system uses RGBD camera, i.e. Kinect in this case, to capture a home base, obstacle(s), and a floor. Then, a free trajectory towards the base is generated. The robot is then commanded via its API to travel along this trajectory. As the robot moves towards the base, the system monitors the home base, the obstacle(s), and the floor and then re-computes the trajectory and override the motion command of the robot. Experimental results are used to demonstrate the performance of the proposed technique.

Manuscript received April 13, 2015; revised October 13, 2015.

II. The Automatic Docking System

The propose system (Fig. 1) consists of an iRobot create (as a mobile robot), Kinect (as a RGBD camera), and a notebook. The notebook connects to the mobile robot and the RGBD camera via USB cables. The camera is powered by a supply from the mobile robot via its expansion port. In this way, the system is very simple and robust. At every servo loop, the control program acquires an RGBD image from the camera, processes the image and extracts the positions of the base, obstacle(s), and floor. Then, the control program computes a free trajectory towards the base and commands the robot to follow this trajectory. The robot's controller has its own processor and is able to control the robot to a desired position. At the next servo loop, the trajectory is updated to get better accuracy as the robot move towards the base.

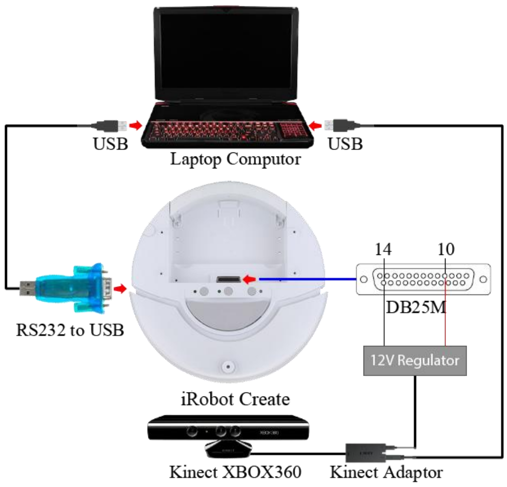


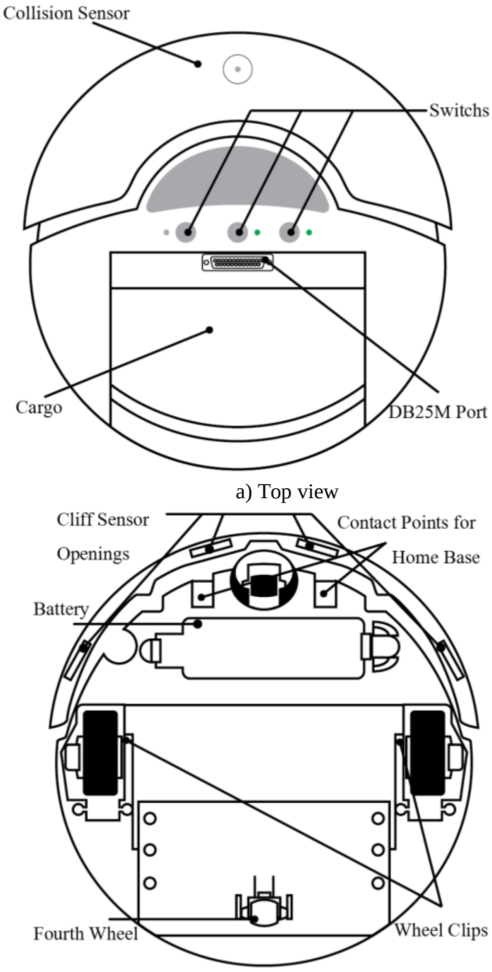
Figure 1. An autonomous docking system.

III. iRobot Create

A "Create" is a commercial mobile robot from iRobot that was introduced in 2007. It is based on their Roomba vacuum cleaning platform which its vacuum mechanism is replaced with a cargo bay and an expansion port. A developer can design a mechanism that fits on a cargo bay that traps the power from the robot and interfaces with the robot controller. This is quite convenience for developers to use the robot as a mobile base and develop new applications. The Create has variety of parts and sensors that can become useful in navigation tasks, such

doi: 10.18178/ijmerr.5.1.11-16

as crash sensors, two 16 bits encoders, LED indicators, buttons and two DC motors (Fig. 2). The maximum speed of the robot is 0.5m/s in both forward and reverse directions. [6] The embedded robot controller is capable of controlling basic functions of the robot including travelling at either a desired distant or speed and handle all the sensor signals. [7]



a) Top view

b) Bottom view

Figure 2. iRobot create.

IV. Kinect and Image Processing Technique

Kinect, launched in 2010, is an interface device for the Microsoft's Xbox 360 video game console. It consists of a RGB camera, a depth camera, a multi-array microphone, and motorized tilt (Fig. 3). The depth camera consists of a laser-based infrared (IR) projector and an IR camera (Fig. 4) which looks for a specific pattern projected onto a scene by the laser-based IR projector. Depth values have been determined by calculating the disparity of each pixel by comparing the appearance of the projected pattern [8]. The camera and the depth sensor is able to output the 8-bit RGB and 11-bit depth images with 640x480 resolutions at the speed up to 30 fps respectively. Both images can be combined and treated as a single image with four dimensions called RGBD (Red, Blue, Green, and Depth) image. The depth image significantly simplifies the segmentation tasks, especially when an object's color is close to an environment. Using Kinect, the object and background with the same color can be easily distinguished by their depths and the RGB image can be processed further to extract the useful information of an object such as the target that is in a predefined pattern of color. There are two limitations for using the Kinect sensor [9]. First, any object with reflective surface may prevent the reflected light from the IR projector from reaching the IR camera, thus causing invalid depth readings at those locations. Secondly, there are blind spot up to approximately 0.4 meters in front of the sensor due to the system requires triangulation between the IR projector and IR camera which mean anything in this range will not be seen by the sensor.

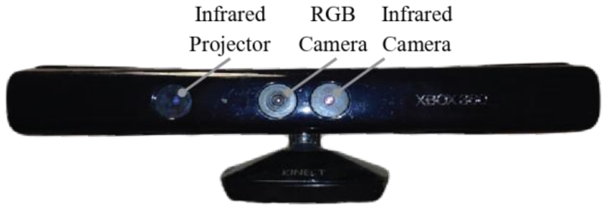
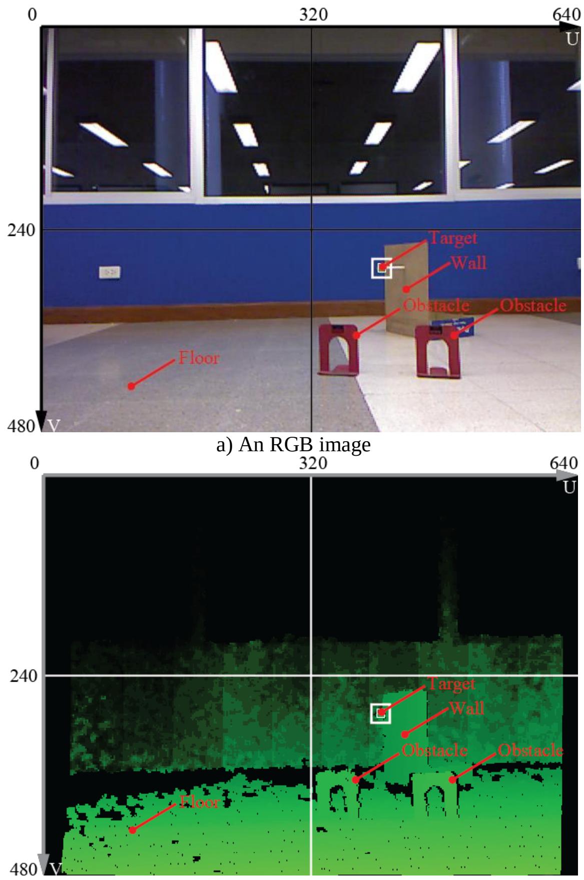


Figure 3. Kinect camera.

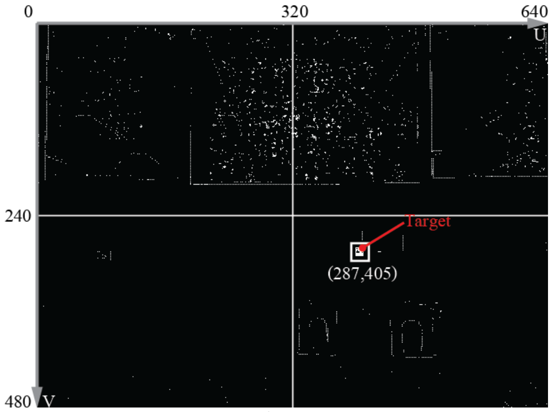


a) An RGB image

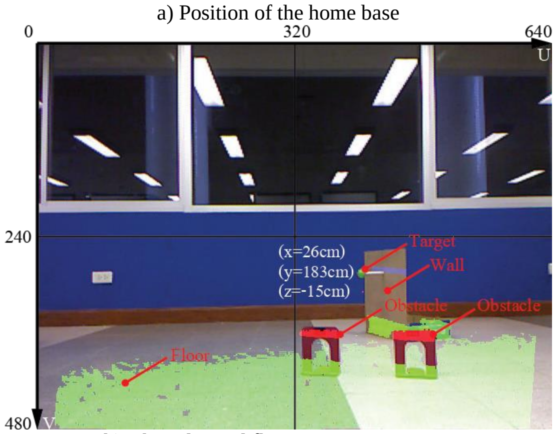
b) A depth image

Figure 4. Raw images from Kinect

The paper proposes a technique to detect a home base, obstacles, wall, and floor and also generate a free trajectory to the home base that is perpendicular to the wall at the base. The onboard depth camera is used to construct a 3D scene [10] which floor and obstacles in the image are distinguished by their height. Point clouds will be identified as wall, floor, obstacle, or ambiguous. The RGB camera is used to find a home base which is a green ball in front of a wall in our case. In this process, RGB format is converted into HSV format [11] which the color can be more precisely identified especially when the light intensity varies [12]. The green ball can then be easily identified by the technique. The position of the green ball in RGB image is then mapped into the depth image [13] to determine the position of the green ball which is the home base (Fig. 5).

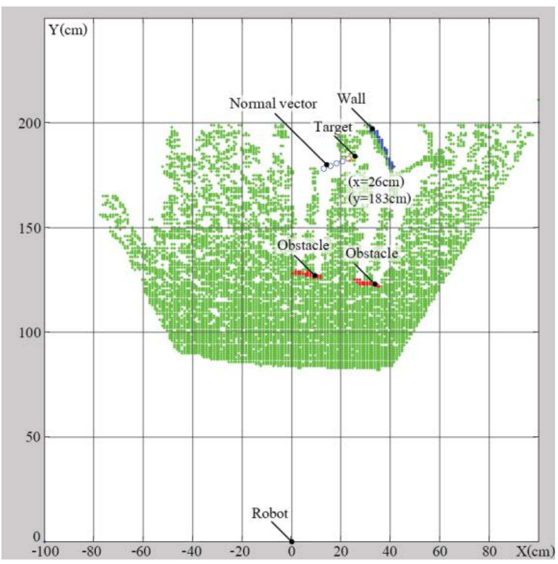


a) Position of the home base

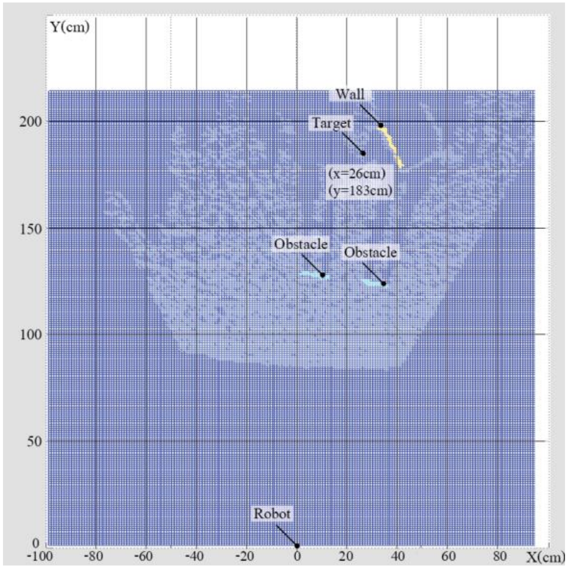


b) Obstacles and floor on an RGB image

Figure 5. Feature extraction from RGBD image.



a) Objects plotted in a 2D map



b) Objects plotted in a grid map

Figure 6. A 2D map with the robot, obstacles, wall, and floor.

V. A Map and Potential Field

A 2D map [14] is built where a robot, a home base, obstacles, and floor are plotted on the map. Occupancy grid map [14] is used to handle all the data (Fig. 6) and objects on the map can be effectively updated. The grid reduces the position resolution and thus the map data and processing time. (Fig. 7) This map is then used to construct the potential field [15]. Roughly, the field will be strong near the obstacles [16], [17] and wall and will be less near the home base. The free trajectory to the base is then generated as a steepest decent trajectory in potential field from the robot to the base (Fig. 8).

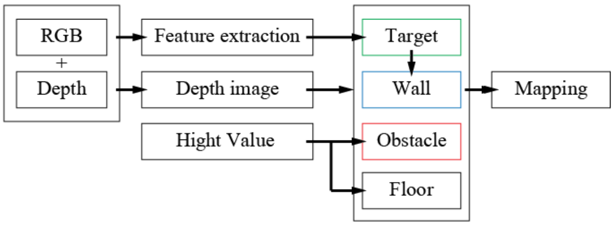
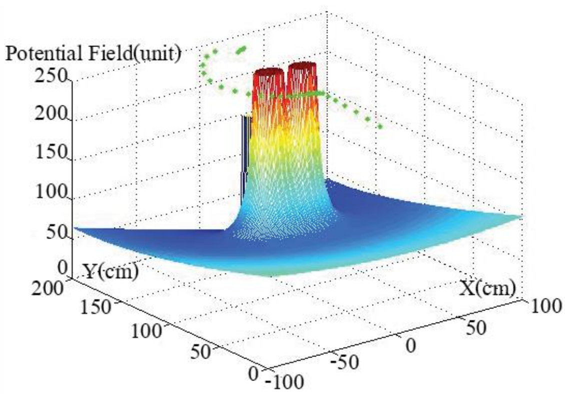
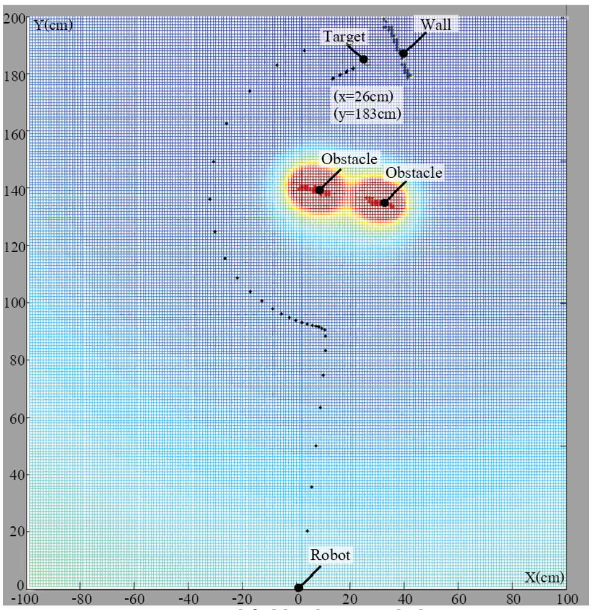


Figure 7. Image processing to construct a map.



a) Potential Field in pictorial view



Potential field in horizontal plane

Figure 8. Potential field.

The accuracy of the technique relies on how accurate we can determine the positions of the base, obstacles and wall. Camera calibration is performed to determine intrinsic parameters of the camera as explained next. Once the camera is calibrated, we can map objects in image space into Cartesian space and vice versa.

VI. Camera Calibration

Photogrammetric calibration [18], Self-calibration [19], and the hybrid technique [18], [20] are all methods used for calibration of cameras. 'Photogrammetric calibration' setup is costly and complex. This process requires a specifically designed 3 dimensional object for calibration by utilizing the actual geometry and form of the object. This will determine the camera model. No object is required when using 'self-calibration' technique. Instead this technique determines the camera model by correlating the number between a static object and a background images at multiple points. This method is not nearly as reliable due to lack of metrics involved. The 'hybrid' [20] technique was proposed by Yudong Zhang. It uses a standard calibration object similar to the plannar chess pattern [21] combined with multiple images of the pattern from different perspectives. This will help determine the camera model. While this method is very reliable, it remains cost effective and is also very flexible.

See Fig. 9 for an example of such chess patterns captured in multiple perspective images. This determines the camera model. Subsequent to calibration, the object calibration positions approximated in Fig. 10.

Intrinsic parameters:

Focal Length: fc = [526.17422 527.21573]

Principal point: cc = [333.43873 261.29703]

Skew: alpha_c = [0.00000]

Distortion: kc = [0.21848 -0.50260 -0.00086 0.00139 0.00000]

Figure 9. Calibration images.

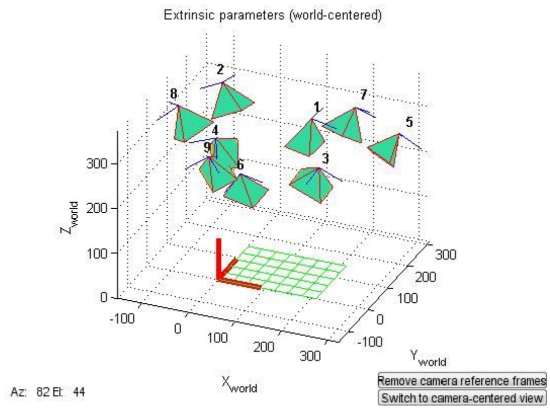
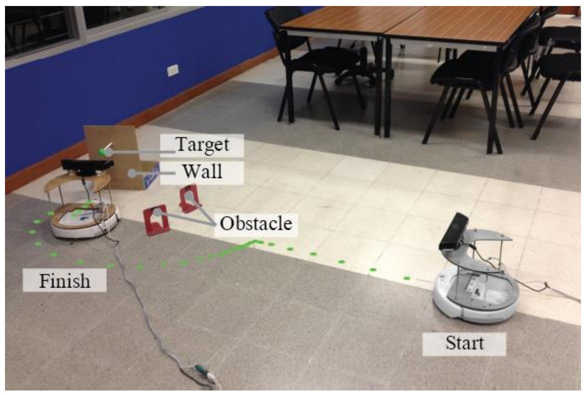
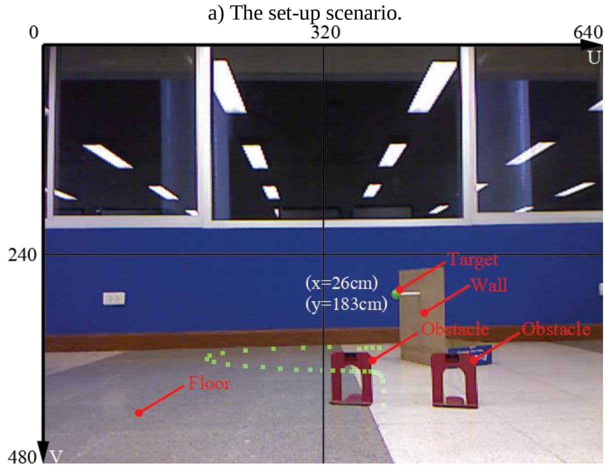


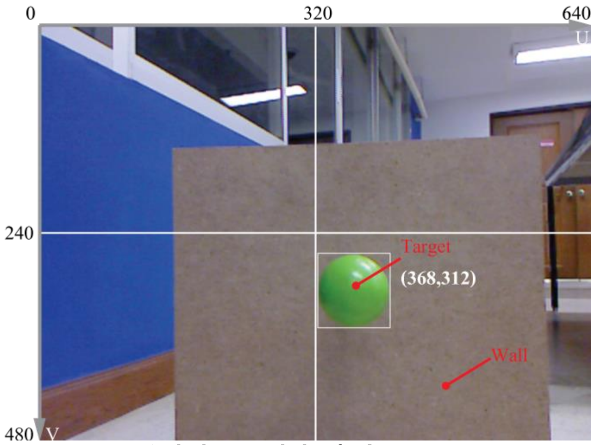
Figure 10. Estimated positions of the calibrated object.



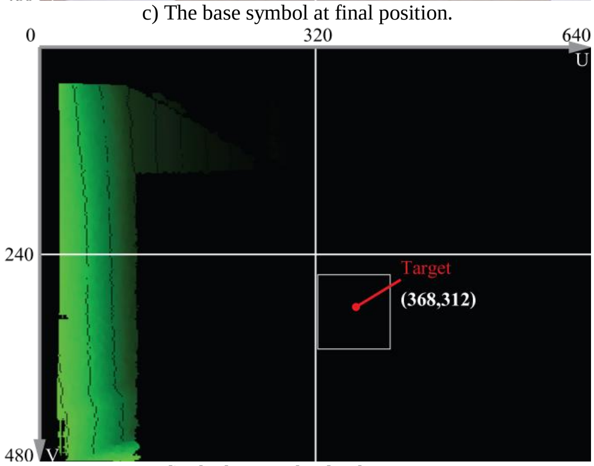
a) The set-up scenario.



b) The robot trajectory in image space.



c) The base symbol at final position.



d) The base in the depth image.

Figure 11. Experimental Result.

VII. Experimental Results

A scenario is set up as seen in Fig. 11. The home base is a green ball in front of the wall. There are two obstacles that block a direct path to the home base. The trajectory should be perpendicular to the wall when docking. The propose system successfully determines the home base, obstacles, and floor and then generates a smooth trajectory to the base that avoids collision with obstacles as seen in Fig. 11a.

The robot is commanded to follow this trajectory via the robot API. The robot embedded processor controls the robot to travel via points along this generated trajectory. At the end, the robot reaches the home base in the direction perpendicular to the wall as desired. The images taken by the onboard RGBD camera are shown in Fig. 11c and Fig. 11d. There is still a positioning error as the green ball is not at the center of the image. At docking position, the depth image cannot track the green ball and the wall behind since they are too close to the camera. Only the position of the base as sensed by the RGB is available in the proposed system. This means that the based is detected but its Cartesian position is not available. We are now further investigating the technique to construct the trajectory based on any available data to improve the positioning accuracy when docking. It is also noted that the robot is a nonholonomic mobile robot that cannot move sideways. The smooth trajectory to dock the robot once the positioning error is detected should be also further investigated. The proposed technique and system can be applied to an automatic parking feature of a vehicle since it has a similar drive mechanism and only RGBD camera and processor are required.

VIII. Summary

An automatic docking with obstacle avoidance is proposed and demonstrated. The RGBD camera with the proposed processing technique can detect the home base, obstacles, wall, and ground. They are then plotted on a 2D grid map. This map with all the mentioned objects is then used to construct a free trajectory that the robot can move to the home base without collision with obstacles. The trajectory is then used to command the robot to travel along this trajectory. The commercial devices, including iRobot Create and Kinect, are used in the study. Experimental result demonstrates the performance of the proposed technique as the robot travel smoothly to the home base.

Acknowledgment

This paper was supported by the Chulalongkorn University Strategic Research Grant CU-57-074-AS

References

[1] R. C. Luo, C. T. Liao, and K. C. Lin, “Vision-based docking for automatic security robot power recharging,” in *Proc. IEEE Workshop on Advanced Robotics and Its Social Impacts*, June 12-15, 2005, pp. 214-219.

[2] B. Tribelhorn and Z. Dodds, “Evaluating the Roomba: A low-cost, ubiquitous platform for robotics research and education,” in *Proc. IEEE International Conference on Robotics and Automation*, April 10-14, 2007, pp. 1393-1399.

[3] M. Bai, Y. Zhuang, and W. Wang, “Stereovision based obstacle detection approach for mobile robot navigation,” in *Proc. International Conference on Intelligent Control and Information Processing*, Aug. 13-15, 2010, pp. 328-333.

[4] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, “RGB-D mapping: Using Kinect-Style depth cameras for dense 3D modeling of indoor environments,” in *Proc. International Conference on 3D Vision*, 2013, pp. 398-405.

[5] G. M. Song, H. Wang, J. Zhang, and T. H. Meng, “Automatic docking system for recharging home surveillance robots,” *IEEE Transactions on Consumer Electronics*, vol. 57, no. 2, pp. 428-435, May 2011.

[6] E. Ruiz, R. Acuna, N. Certad, A. Terrones, M. E. Cabrera, “Development of a control platform for the mobile robot roomba using ROS and a Kinect sensor,” in *Proc. Latin American Robotics Symposium and Competition*, Oct. 21-27, 2013, pp. 55-60.

[7] iRobot® Create Owner’s Guide

[8] B. Peasley and S. Birchfield, “Real-time obstacle detection and avoidance in the presence of specular surfaces using an active 3D sensor,” in *Proc. IEEE Workshop on Robot Vision*, Jan. 15-17, 2013, pp. 197-202.

[9] W. Zeng and Z. Zhang, “Microsoft Kinect sensor and its effect,” *IEEE Computer Society*, IEEE MultiMedia, April-June 2012

[10] J. Cunha, E. Pedrosa, C. Cruz, A. J. R. Neves, and N. Lau, “Using a depth camera for indoor robot localization and navigation,” in *Proc. Robotics Science and Systems (RSS) RGB-D Workshop*, June 2011.

[11] M. Sezgi and B. Sankur, “Survey over image thresholding techniques and quantitative performance evaluation,” *Journal of Electronic Imaging*, vol. 13, no. 1, pp. 146–165, 2004.

[12] N. M. Ali, N. Khair, A. Md Rashid, and Y. M. Mustafah, “Performance comparison between RGB and HSV color

segmentations for road signs detection," *Mechanics and Materials*, vol. 393, pp. 550-555, 2013.
[13] A. Ming and H. Ma, "A blob detector in color images," in *Proc. 6th ACM International Conference on Image and Video Retrieval*, New York, USA, 2007.
[14] R. Mojtahedzadeh, "Robot obstacle avoidance using the Kinect," Master dissertation, KTH Computer Science and Communication, Stockholm, Sweden, 2011.
[15] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Proc. IEEE International Conference on Robotics and Automation*, Mar. 1985, vol. 2, pp. 500-505.
[16] M. A. Goodrich, Potential Field Tutorial.
[17] Y. Koren and J. Borenstein, "Potential field methods and their inherent limitations for mobile robot navigation," in *Proc. IEEE International Conference on Robotics and Automation*, Apr. 9-11, 1991, vol. 2, pp. 1398-1404.
[18] F. Remondino and C. Fraser, "Digital camera calibration methods: considerations and comparisons," *Image Engineering and Vision Metrology*, 2006.
[19] C. S. Fraser, "Invited review paper digital camera selfcalibration," *ISPRS Journal of Photogrammetry & Remote Sensing*, vol. 52, pp. 149-159, 1997.
[20] Z. Y. Zhang, "Flexible camera calibration by viewing a plane from unknown orientations," in *Proc. Seventh IEEE International Conference on Computer Vision*, 1999, vol. 1, pp. 666-673.
[21] W. Poomarin, K Chuengsatiansup, and R. Chanchareon, "Visual positioning of a delta robot," in *Proc. 3rd IASTED Asian Conference on Modelling, Identification, and Control*, AsiaMIC, 2013, pp. 227-232.

Warin Poomarin received the B.Sc. Degree in Physic from the Chiang Mai University, Chiang Mai, Thailand, in 2010.
Currently, he is active Master Degree Student, Department of Mechanical Engineering, Chulalongkorn University. His research works are in the area of mechatronic and robotics systems. He is also a freelancer Visual Effect Artist specializing 3D modeling and dynamic simulation base in physic.

.

Dr. Ratchatin Chancharoen received his BS degree in mechanical engineering from Chulalongkorn University in 1991, MS degree in mechanical engineering from Oregon State University in 1994, and PhD degree in mechanical engineering from Chulalongkorn University in 2000.
He is currently an Associate Professor at the Mechanical Engineering Department, Chulalongkorn University, Thailand.

Dr. Viboon Sangveraphunsiri received the B.Eng. degree in Mechanical Engineering from Chulalongkorn University in 1978, and received MS.ME. and Ph.D. degrees in Mechanical Engineering from Georgia Institute of Technology, Atlanta, U.S.A., in 1980 and 1984 respectively.
He is the Professor of Mechanical Engineering, Chulalongkorn University. Dr. Sangveraphunsiri works in multiple areas of Dynamic Control and Robotics: advanced control systems, force control, computer vision, manipulators, autonomous mobile robots, CAD/CAM/Rapid-prototype in product design and manufacturing. Since 2008, he has founded the Regional Center of Robotics Technology, Faculty of Engineering, to be a research center in the areas of Control and Robotics at Chulalongkorn University. The research works of the center are more toward medical robots as well as industrial applications.