Design of Map Decomposition and Wiimote-based Localization for Vacuuming Robots

Chia-Pin Kuo, Shyh-In Hwang
Department of Computer Science and Engineering
Yuan Ze University
s956007@mail.yzu.edu.tw, shyhin@cs.yzu.edu.tw

Abstract-*Vacuuming robots have been more and more popular in the house cleaning. But most of such products on the market do not work efficiently. The common drawbacks are random cleaning path, lack of map information, and excessive power consumption. To solve the problems, a localization system is needed. The techniques such as Zigbee, wireless or ultrasound were adopted in the literature. However, these methods suffer from high setup cost and large position error. This paper proposes a creative projection localization system equipped with a Wiimote controller and a digital compass. In addition, we apply a Boustrophedon algorithm (Back and forth motion), and a map decomposition algorithm to determine a better cleaning path. By implementing the robotic system, we not only improve the vacuuming robot's efficiency but also prove the practical use in robotic localization.*

Keyword: Vacuuming Robot, Wiimote, Boustrophedon, Digital Compass, Map Decomposition.

1. Introduction

Intelligent robot developments have become more and more popular in recent years. Many robotic researchers are trying to create smarter robots and let them helping people in our life. The famous one is house cleaning robot. For example, there are iRobot's Roomba, Electrolux's Trilobite, Yujin Robotics' i-Clebo, and LG's V-R4000, etc. Common features they have are scheduled to clean periodically, dodge obstacles, auto-recharge, and detect floors. If people have busy works every day, vacuuming robot must be a good helper.

Unfortunately, there are some drawbacks when the robot works. Take Roomba for example, one of these is it cannot plan cleaning path. It starts with spiral circles and then takes a random direction to clean. On this circumstance, the robot is likely to waste time cleaning the same areas or miss some regions. In addition, excessive power consumptions may be happened.

The paper will describe a low cost, accurate, and creative localization solution to assist vacuuming robot in the housecleaning. Taking costs and programmability as considerations; we choose iRobot's Roomba for the robot platform. The main idea is using Wiimote's camera to capture IR LED set up on the robot and support position data. More details will be described in the 3rd sections below.

2. Related Work

Indoor localization must be precision and more stable than outdoor system. But nowadays systems' error ranges are still about fewer meters. The common techniques are such as follows: ultrasound [4], infrared (IR) [1], wireless [5], Zigbee [7], and RFID [3]. Expensive set up costs are their current features.

In the map system, CMU's Choset had proposed a cellular decomposition [2] technique and let the robot take Boustrophedon motion in the decomposed cell; Zack J. Butler also yielded a rectilinear decomposition [8] for robot working in special platform.

Automatic projector calibration is a popular application. Through calibrating steps can gain more accurate coordinates. This kind of technique is useful in electronic white board, multi-touch, and other multi-media programs. Rahul Sukthankar proposed camera-based computer vision techniques [6] for surface projecting. The method is using the homography matrix to correct the different images.

3. System Overview

Our system includes robot controlling modules and some applications, as shown in Figure 1, the architecture below demonstrate VIPER embedded board connects Roomba and compass through I/O ports.

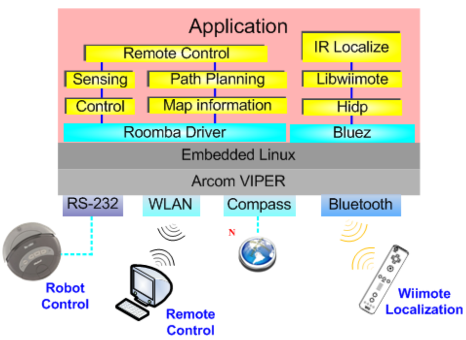


Figure 1. System architecture

3.1 Hardware introduction

Whole system bundles many components no matter hardwire or software. In the brginning, we choose the Roomba Discovery as a main platform, because it not only offers SCI document for programming but also embeds many functions, such as bounding cleaning, virtual wall, and auto-recharge. Second, the VIPER embedded board, as shown in Figure 2; all programs will be cross-compiled and downloaded on it.

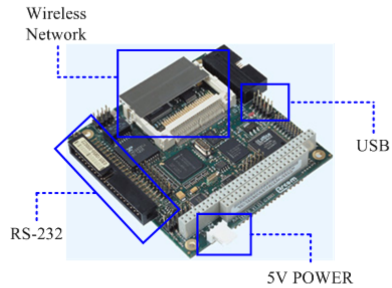


Figure 2. VIPER system board

The next is digital compass which can assist robot to correct its own directions. In order to gain the compass' data easily, we use Basic Stamp 2 (BS2) IC to connect with it. The Figure 3 below shows the Parallax BS2 IC combines the Hitachi HM55B digital compass.

Figure 3. BS2 IC and digital compass

The finally is Wiimote game comptroller. It is produced by Nintendo Company Ltd. Embedding 3-axis accelerator chip and IR camera inside. For the reason, Wiimote can sense user's swing motion and control game program. Our paper will use its IR camera to capture IR

LED sources equipped on the Roomba, as shown in Figure 4; the IR camera (Red circle) is in front of the controller. Depending on this tracing relationship, we can know where the robot is easily.

Figure 4. Wiimote and IR sensor bar

3.2 Development software

In order to implement our system, we also need some application tools to build them. For example, VIPER embedded board offer its cross-compiler environment named "arm-linux-gcc 3.4.x". So we will use this tool to burn our C code into the board. In other hands, Roomba and Wiimote release their free API for programming, just like "Serial communication interface (SCI)" and "Libwiimote-0.4". Finally, when analyzing Bluetooth packets, System must be installed "Hidp", "Bluez", and "Bluez-libs-devel" these packages on it.

4. Using the Homography

The Wiimote-based localization system will use image homography process. The main idea is to find out the homography matrix between source image and target plane, as shown in Figure 5, the mathematics matrix can warp the IR sources pixels to fit the projection plane.

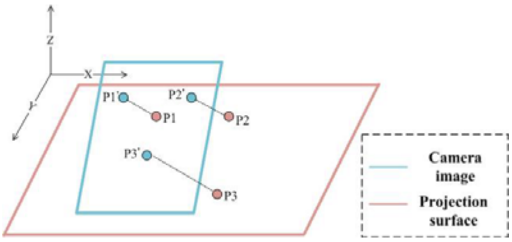


Figure 5. Homography projection

The Figure 5's math formation is:

$$p' = Hp$$

$$=> \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$=> \quad x' = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + h_{33}}, \quad y' = \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + h_{33}}$$

In order to figure out the matrix value, we need at least four points in the plane. Though points' calibration, Wiimote camera will map the IR pixels to the projection ones. If the projector is flipped upside down or rotated, the homography matrix will correct the target surface pixels automatically.

When homography process is completed, projection pixels must be translated to the world distances. Because slopping projection will lead to the trapezoidal image, as shown in Figure 6, here we will use bilinear interpolation to solve the geometric problem.

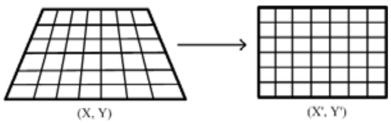


Figure 6. Geometric transformation

The transformer formation is on the following:

$$x' = a_0 + a_1 x + a_2 y + a_3 xy$$

$$y' = b_0 + b_1 x + b_2 y + b_3 xy$$

5. Map Creation and Path Planning

Our paper here will introduce a map system. How does it store the data, create, decompose and merge for each other? In addition, the vacuuming robot can take some path strategies to fit the different situations. All of the details will be described below.

5.1 Rectilinear map

In order to demonstrate the robot's "back and forth" motion clearly. We choose the "rectilinear" map to implement. Assuming that map and obstacles' outline are all rectangles. System will record objects' "top-left" and "bottom-right" coordinates data, as shown in Figure 7.

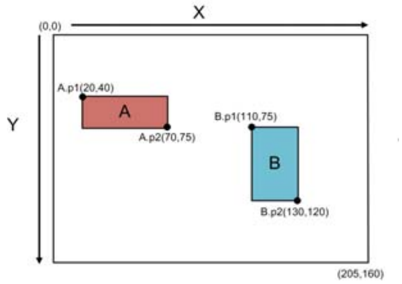


Figure 7. Recliner map expression

5.1.1 Map decomposition

Our map system will decompose and merge by some rectangle obstacles. System first finds out blocks' "forward" and "backward" edges and then cuts along with these sides. Now we can get raw regions called "cell", as shown in Figure 8, map is decomposed for ten cells by A, B, and C blocks.

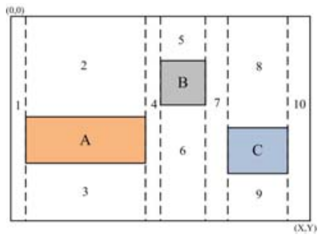


Figure 8. Decomposed cleaning cells

Then these cells are able to search for each other and check if need to be merged. If the cell width is less than the Roomba's size, such as cell 1, 4, 7, and 10 above, they will be combined by the neighboring bigger cells. When the map creation process is completed, system will generate final cells for cleaning. These cells are called "target cell". They can be seen in Figure 9, the white spaces are the missing ones Roomba can not pass.

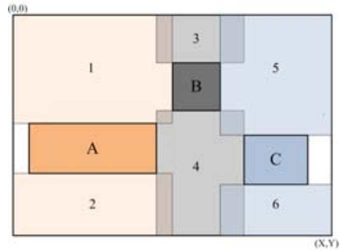


Figure 9. The Final cleaning cells

5.2 Boustrophedon motion

The Boustrophedon path is most likely the cow sowing in back and forth motion, as demonstrated in Figure 10. The path's feature is simple and no repeated. For this reason, robot can clean faster and more efficiently. In order to let the Roomba walk like that, we also need the digital compass to correct its forward direction.

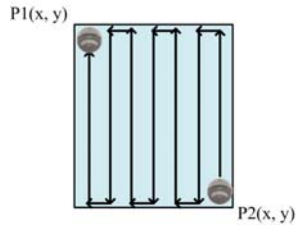


Figure 10. Boustrophedon motion

5.3 Path planning

Ones robot has received working instructions; it will start to go the target cell and clean. But in other situations, robot may take different strategies to fit them for example contacts obstacles when exploring.

5.3.1 To find the starting point

If the Roomba confirm where the cleaning area is, it will go to one of the nearest corners for the cell and then stand by. We can classify the follow situation: "in" or "out" the cell, as shown in Figure 11, robot first will walk by y axis and then x axis.

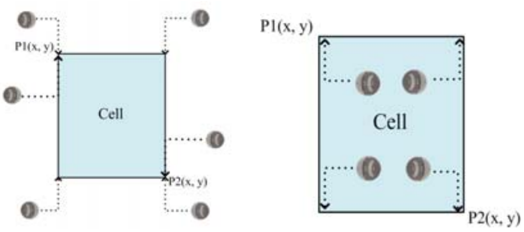


Figure 11. out of the cell (left); in the cell (right)

5.3.2 Dodging planning

It is possible that robot comes across obstacles when cleaning. Therefore it take dodging strategy to walk along by the blocks with position assistance. Robot will compare its original location with target point, and then chose the shortest path to dodge along, as shown in Figure 12.

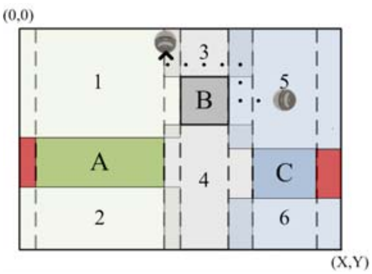


Figure 12. Roomba dodges along block B

When the robot is in the target cell successfully, it will continue to clean in Boustrophedon motion, as shown in Figure 13.

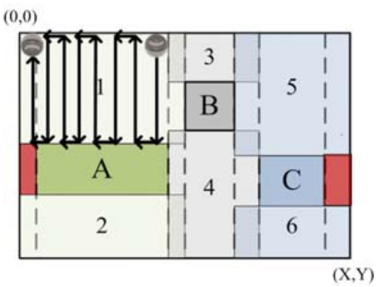


Figure 13. Boustrophedon in the target area

6. Implementation

This section here will describe the implementation and results.

6.1 Environment introduction

The Wiimote is set up in the high stick tall about 180 cm. The front camera is face to the ground sloppily with 60 degrees angle, as shown in Figure 14.

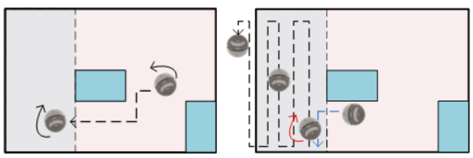
Figure 14. Wiimote controller setting up

Finally, the testing range robot will clean is about 205 x 160 (cm), whole localization environment is demonstrated in Figure 15.

Figure 15. System overview

6.2 Motion testing

The map's scenario is with two blocks in it. The robot's dodging and Boustrophedon path is as Figure 16 (a) (b) shown.



(a) Dodging (b) Boustrophedon

Figure 16. Path testing; (a), (b)

All of the above testing videos are on the following links:

a. http://www.mmlab.cse.yzu.edu.tw/~davian/

demo_1.rar

b. http://www.mmlab.cse.yzu.edu.tw/~davian/demo_2.rar

7. Conclusion

Our paper is successful in implementing an accurate localization system with a Wiimote game controller. After geometric transformation, the error range can be limited in as small as 2 cm. Furthermore, while a Wiimote controller is cheape (less than one thousand NT dollars) and easy to handle, everyone can use it at home conveniently.

To summary, our research achieves the following goals:

- Using an embedded board, and some sensor devices to control the popular Roomba robot successfully.
- Integrating a Wiimote controller to facilitate the calculation of position data.
- Designing map decomposition algorithm to improve the cleaning path.
- Implementing the Boustrophedon motion to let the robot move in a regular back-and-forth path and using digital compass to correct its moving directions.
- Vacuuming robots can adopt some path strategies to dodge obstacles.

Our system can still be improved in the following directions:

- Using gyro sensor to replace the digital compass to avoid unnecessary magnetic disturbance.
- Implementing a multi-robot system for better cooperation.
- Improving the map system to fit the real environments in our life.

Reference

[1] A. Harter and A. Hopper, “A distributed location system for the active office,” IEEE Network, vol. 8, no.1 January/February 1994, pp. 62 – 70.

[2] Howie Choset, “Coverage of Known Spaces: The Boustrophedon Cellular Decomposition,” IEEE Autonomous Robots, Vol.9, No.3, pp.247-253, 2000.

[3] Lionel M. Ni, Yunhao Liu, Yiu Cho Lau and Abhishek P. Patil, “LANDMARC: Indoor Location Sensing Using Active RFID,” First IEEE International Conference on Pervasive Computing and Communications (PerCom'03) March 23 - 26, 2003.

[4] Nissanka B. Priyantha, Anit Chakraborty and Hari Balakrishnan, “The Cricket Location-Support system,” Proc. 6th ACM MOBICOM, Boston, MA, August 2000.

[5] P. Bahl and V. Padmanabhan, “RADAR: An In-Building RF-Based User Location and Tracking System,” Proc. IEEE INFOCOMM 2000, Mar. 2000.

[6] Rahul Sukthankar, Robert G. Stockton, Matthew D. Mullin, “Smarter Presentations: Exploiting Homography in Camera-Projector Systems,” Proceedings of International Conference on Computer Vision, 2001.

[7] Stefan Schwarzer, Martin Vossiek, Markus Pichler and Andreas Stelzer, “Precise Distance Measurement with IEEE 802.15.4 (ZigBee) Devices,” Radio and Wireless Symposium, 2008 IEEE.

[8] Zack J. Butler, “Contact sensor-based Coverage of Rectilinear Environments,” Proceedings of the 1999 IEEE International Symposium on.