

Evaluating Different Landmark Positioning Systems within the RIDE Architecture

Joaquín López, Christopher Watkins, Diego Pérez and Miguel Díaz-Cacho

Abstract—Mobile robots operating in the real world need a very reliable localization system to navigate autonomously for long periods of time. Numerous methods for indoor mobile robot localization have been developed. However, an affordable system covering all environments and situations is not yet available. Therefore, it is very important for mobile robot application developers to be aware of the operation and limitations of the different localization systems in order to obtain the best performance for each case. This paper evaluates two indoor localization systems that are integrated in the RIDE architecture: a commercial (Hagisonic StarGazer) and a low cost localization system based on the popular Wii remote control (WiiMote) with different tag distributions were evaluated. Characteristics that were tested include precision, accuracy, reliability, cost and immunity to interference.

Index Terms—Mobile robot localization, control architecture, landmark localization system.

I. INTRODUCTION

ROBOT localization is the problem of determining a robot's position in a known environment, assuming that it is provided with a map of the environment. Mobile robot localization is essential to autonomously navigate within the working area. Numerous methods for indoor mobile robot localization have already been developed [1], [2]. These techniques have been categorized by different authors as relative positioning and absolute positioning [3]. Still, research on this subject continues because a low cost system that covers all environments and situations has not been yet designed.

The position of a robot can be obtained using relative positioning techniques with sensors that measure the movement such as encoders, gyroscopes and accelerometers. However, these kinds of techniques by themselves are subject to cumulative errors.

The absolute positioning techniques rely on a map of some kind of landmarks and sensors that can detect the position and orientation of the landmarks close to the robot while navigating on the different areas of the map.

Most of the localization systems include relative position measurements and absolute relative measurements. Both kinds of information are integrated using some statistical filter such as a Particle filter [4] or Kalman filter [5]. Particle filters [4],[6] represent the belief distribution by a set of N random samples or particles while Kalman filters assume a normal distribution. In both cases it is crucial to correctly characterize the motion

and measurement models. The goal of this paper is to extend the analysis in [7] about two infrared landmark measurement systems and provide useful hints for researchers who would like to use them.

The rest of the paper is organized as follows. The next section introduces the localization systems. Sections III and IV describe the integration in the RIDE architecture [8] and the new control modules. The results of the different tests, for both the Hagisonic StarGazer and the WiiMote based system, are presented in section V. Finally, section VI presents an alternative localization system with a different tag distribution.

II. THE LOCALIZATION SYSTEMS

In both localization systems considered here the infrared (IR) landmark positions are provided as the map of the working area. The landmarks are retroreflective tags located on the ceiling of the indoor places where the robot navigates. An IR camera pointing upwards detects the relative landmark positions while the robot navigates. The robot location can be obtained when at least one landmark is detected within the IR camera field of view. Dead-reckoning information can also be integrated using some of the multiple statistic solutions proposed in this field such as a particle or Kalman filters.

Dynamic environments can include different kinds of obstacles that can change the position such as people and objects moved by people. If the tags were located on the walls, these obstacles could occlude the tags. However this problem does not occur with the location proposed by the two different systems studied in this paper. It can be therefore a good solution for applications where the environment includes a lot of moving obstacles such as tour guide robots in museums or fairs where other systems that rely on occupancy grid maps [9], [10] might fail.

A. The Hagisonic StarGazer Robot Localization System

This device (figure 1) consists of a camera that detects infrared radiation reflection from tags that are placed on the ceiling of a room. The IR radiation comes from LEDs that are located around the camera. The captured images are analyzed and the system returns the position of the robot and the angular heading with regards to the landmark.

The format of the landmarks is given in figure 2, where the white dots (IR tags) and empty corner are common to all landmarks to allow the StarGazer to calculate heading angle. All possible binary combinations of the red dots (tag/no tag) form the number of different landmark ID's possible ($2^5 = 32$). The frame of reference used to specify the position

Joaquín López, Diego Pérez and Miguel Díaz-Cacho is with University of Vigo. E-mail: joaquin, dplosada, mcacho@uvigo.es

Christopher Watkins is with University of Melbourne. E-mail: chris.watkins88@gmail.com



Fig. 1. StarGazer device with the camera and the array of led pointing in the same direction.

is defined as shown in the figure, with the origin at the center of the white dot in the corner opposite the empty corner. When viewed from above, a clockwise rotation of the StarGazer results in an increase in the heading angle, ω . If the landmarks are set up in such a way that they line up with North, the heading angle will correspond to a compass with 90 being East and -90 being West.

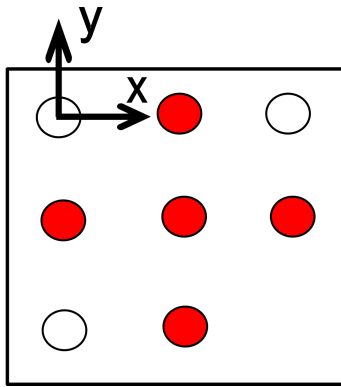


Fig. 2. Landmark format. Bird's eye view from above the ceiling.

The manufacturer claims the following specifications:

- UART hardware interface (TTL 3.3V) 115.200bps
- Size: 50x50x28 mm
- Location range (per tag) 2.5 to 5 m in diameter
- Precision of 2 cm
- Heading Angle Resolution 10°
- Power consumption: 5V 300mA, 12V 70mA.

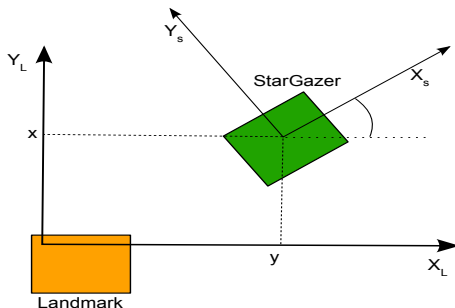


Fig. 3. StarGazer and a single landmark. Bird's eye view from above the ceiling.

The StarGazer device has two modes of normal operation (alone and map). In alone mode the device continually reports the ID, heading angle and relative position with respect to any visible landmark. In map mode the device is able to itself create the map of tags and obtain the robot position in the map. However, for this research only the alone mode was considered.



Fig. 4. The WiiMote based localization device has an array of leds pointing in the same direction as the WiiMote.

While calculating position the StarGazer outputs data on the serial connection in the format:

<Mode, ID, Angle [deg], X [cm], Y [cm]>

Where the *Angle*, *x*, and *y* define the sensor position relative to the landmark. When there are no landmarks within view, a message saying “DeadZone” is sent and followed by silence until a landmark is found again.

B. The WiiMote-based Localization System (WLS)

The Wii remote control known as WiiMote includes an IR camera with a resolution of 1024x768 pixels sensitive only to bright sources of IR light. The WiiMote can track up to four IR light sources at 100Hz [11] and is used in game applications together with a led bar either above or below the screen.

The system has been used for mobile robot localization with different configurations. For example, in [12] the Wii remote controller is placed on the mobile robot pointing upward and several IR leds are placed on the ceiling. However, equipping buildings with active LED emitters can be difficult due to the power and maintenance requirements of the LEDs. One way to overcome this problem is to use retroreflective tags as landmarks like the tags used by the Hagisonic StarGazer. In this case, these tags will reflect the IR light produced by an array of LEDs located on the robot. The IR light source is with the tracking camera rather than the tracked point that only reflects light. Figure 4 shows this solution that has been presented in [13].

The landmark positions are provided as the map of the working area. The robot location can be obtained when at least two landmarks are detected within the IR camera field

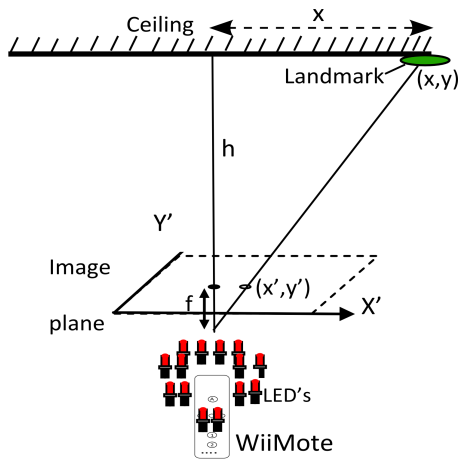


Fig. 5. WiiMote and illumination LED's pointing to the ceiling where the tags are attached.

of view. By using this device we reduce the price of the final localization system still obtaining an excellent performance in most indoor environments.

The tag coordinates returned by the WiiMote are the pixel coordinates in the IR camera (x', y') . A conversion from pixel to metric distance is needed.

Relation (1) can be observed from figure 5.

$$\frac{x'}{f} = \frac{x}{h} \quad (1)$$

Where x' is the distance in pixels, x is the corresponding metric distance at the ceiling, f is the image focal distance and h is the distance from the WiiMote to the ceiling. The metric distance is then:

$$x = \frac{h \cdot x'}{f} = p \cdot h \cdot x' \quad (2)$$

In order to compare the performance of both systems, it is necessary to build landmarks out of several retroreflective tags. However, the WiiMote will track only four IR sources. Since the white dots at the corner are common to all landmarks to allow calculate the heading angle, there is only the possibility to add a new tag in one of the five positions. Even though, the number of possible IDs in this case is 6, in order to make the ID detection algorithm more robust, all the landmarks need to have four tags. Any reading from the WiiMote that provides less than four tags is discarded.

III. INTEGRATING THE IR LOCALIZATION SYSTEM IN THE ROBOT CONTROL ARCHITECTURE

A. The control architecture

The mobile robot control architecture, based on RIDE, is a modular control architecture that is organized as shown in figure 6. Even though the different modules are organized in four sets, they can be mapped to the three layer architecture popularized by Bonasso et al. [14]. The hardware servers and control set implement the functional layer while Task Dispatch implements the executive and planning layer. Finally RIDE

includes a set of processes to interact with the users and connect to other process for Multirobot applications.

The navigation platform is based on CARMEN [15] and some modules such as localize, navigator and base hardware servers remain basically the same. Unlike CARMEN, motion control is divided into high-level (strategic) planning [16] and lower-level (tactical) collision avoidance using the Beam method [17].

The hardware server modules govern hardware interaction providing an abstract set of actuator and sensor interfaces and isolating the control methods from the hardware details. Most of the hardware devices are connected to a CAN bus using RoboCAN [18]. Some of these devices are used in navigation such as the laser and sonar while others are specific for the application such as the robot head, sound and speech system, etc. The hardware servers also provide low-level control loops for rotation and translation velocities. Thanks to this layer changes in hardware components can be made without changes on higher layers modules while keeping the same interface.

The control modules integrate sensor and motion information to provide improved sensor odometry, basic navigation capabilities (localization, path planning, follow path, etc) and basic application specific functions (say text, make expression, etc).

All the modules in the executive layer belong to the RoboGraph application that includes two modules (figure 8): task editor that is used only for application development and task dispatch without graphical interface that should be working when the robot is executing a task.

This layer uses hierarchical interpreted binary Petri nets [19] to coordinate the activity of all the rest of the modules. Tasks are described using an interpreted Petri net editor and saved in an XML file. A dispatcher loads these files and executes the different Petri nets under user requests. A monitor that shows the state of all the running nets is very useful for debugging and tracing purposes.

The interaction with other modules in the architecture is done through publishing and subscribing to messages. This way, problems on a module such as a blocking problem do not block dispatch and we can set up simple mechanisms to detect and recover from a failure or exception situation.

The Petri Net can evolve only with the arrival of messages or the end of a Timer. Every time a change in the status of a Petri net (start, stop, evolve) or in the waiting queues (new requests added or removed) is produced, a new message reporting that change is issued for GUI monitor mode and stored in the log file for GUI play-logger mode. There are several interface modules for the programmer to debug and trace the control and hardware server modules. However, there is only one interface module on board that allows the user to interact with the robot (for example, for identification). Finally, users can also connect via Web, monitor and interact with the robot.

Each module in figure 6 is a Linux process that exchanges information with other modules using IPC Inter Process Communication. Developed at Carnegie Mellon's Robotics Institute, IPC provides a publication-subscription model for processes to pass messages to each other via a central server process. Each application registers with the central server, and

specifies what types of messages it publishes and what types it listens for. Any message that is passed to the central server is immediately copied to all other processes subscribed.

The process of building a mobile robot application using this framework includes programming on different levels. First, hardware server and control modules need to be implemented. Modules that implement navigation tasks can be used from one application to another. At the next level, the executive layer, it is necessary to build a module or sequencer that sends requests and receives the outcomes of the functional layer modules. This module usually varies from one application to another.

B. The localization module

The former localization module included in RIDE was the CARMEN module [15] that implements a variation of Monte-Carlo Localization [20] algorithm. The module accepts odometry readings and laser readings from the laser module. One goal of this research is to provide RIDE with another two different localization systems. For that purpose, we follow the good design practices proposed in [15] implementing in different modules the sensor interfaces and the localization method. The sensor interfaces developed here are the modules StarGazer and WiiMote described in the next section.

Localize module is able to integrate odometry and information obtained from the laser, StarGazer or WiiMote to keep track of the robot localization. When starting, it subscribes to one or several sensor information messages according to the robot configuration. The robot position can be represented by particles or a by a normal distribution, depending on the integration process (particle filter or Kalman filter). Every time a new sensor readings message arrives, a new update position step is executed. It is also possible to interleave update position steps of different kind of sensors (laser, StarGazer or WiiMote) since all use the same position distribution.

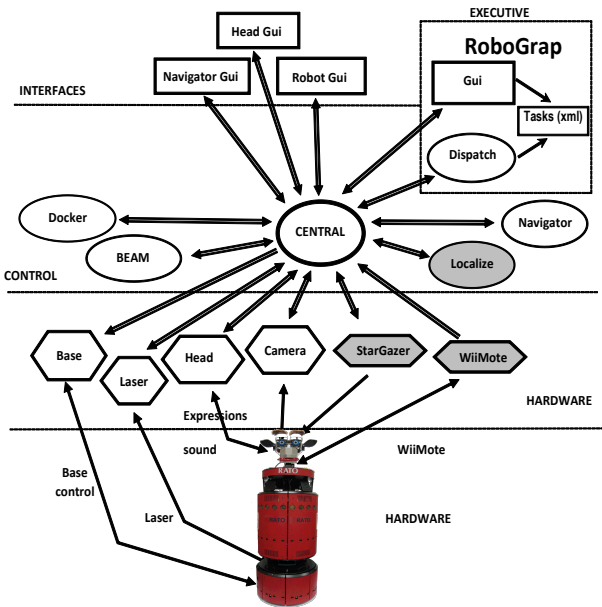


Fig. 6. Robot control architecture. Stargazer and WiiMote modules are the hardware drivers that control the devices used in this work. Localize subscribes to IPC WiiMote and Stargazer messages.

IV. LOCALIZATION DEVICE CONTROL MODULES

A. StarGazer control module

The StarGazer control module includes two threads as shown in figure 7. The first thread handles the communication with other software modules via IPC while the second thread handles the communication with the StarGazer device using the serial port.

The serial reading thread simply waits for available input on the serial connection. When data becomes available it reads it and stores it into the semaphore protected buffer queue which is a global variable shared between the threads.

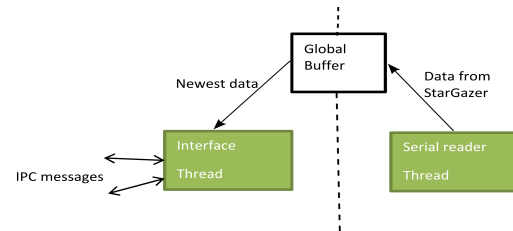


Fig. 7. The StarGazer module includes two threads that take care simultaneously of IPC and device events.

The interface thread listens for IPC messages. Namely, configuration messages that indicate to change the module configuration and odometry messages. When an odometry message is received this module publishes a new message with the odometry and StarGazer data combined. The StarGazer data is the newest stored in the buffer queue.

The IPC messages that define the interface with the rest of the modules are very similar to the Laser messages used by localize.

B. WiiMote control module

The Wii Remote is a wireless device, using standard Bluetooth technology to communicate with the Wii. It is built around a Broadcom BCM2042 bluetooth System-on-a-chip, and contains multiple peripherals that provide data to it. As such, it will appear as a standard input device to any Bluetooth host and we use this communication system to transfer data between the WiiMote and the robot on-board computer.

The WiiMote module was programmed using the Cwiid library [21] that installs a callback function executed every time a new reading from the WiiMote is obtained. If necessary, a new IPC message is created and sent.

The WiiMote also contains a $\pm 3g$, 8-bit, 3-axis accelerometer operating at $100Hz$ and a set of buttons. The IPC message published by the WiiMote module includes information of all these devices and it is used by other modules besides localize such as the BEAM module. When the BEAM module is in wiiControlled mode, it uses the button information of the WiiMote message as the base movement reference. This way, the WiiMote can be removed from the localization device and used as a simple remote control.

The localize module gets the tags info (pixel positions) from the WiiMote message, obtains the tags position according to equation (2) and obtains the landmark position and orientation.

At this time the information is the same as the information provided by the StarGazer sensor.

V. EVALUATING AND CHARACTERIZING THE LOCALIZATION SYSTEM

The testing of the StarGazer undertaken here was done in order to quantify the operating characteristics and performance of the device. Relevant specifications in the area of mobile robot localization include repeatability and accuracy of measurements, effect of interference and localization range per landmark. This data will enable better decisions to be made about suitable environments and applications for the device. It will also provide a benchmark with which other systems can be compared.

A. StarGazer System

The tests were performed with the StarGazer device sitting on a table and one or two landmarks located at different positions (figure 8). Each test was performed at heights of approximately 2.5m, 3m and 3.5m. Most of the distances were measured using a Leica DISTO D2 Laser Distance Measuring tool. For static values, a collection of about 700 samples were obtained for each point to obtain the average value.

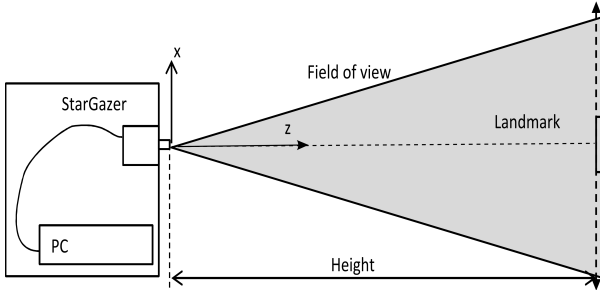


Fig. 8. Tests set up to evaluate the system.

Various tests were created to measure the operating characteristics of the StarGazer:

Height Test: The robot working area of some applications such as a tour guide robot might have ceilings with different heights. It is therefore important to know the limitations of the localization systems according to this parameter.

Landmark type HLDx-1 was usable for heights over the range 2200mm to 4300mm and HLDx-2 worked from 3300mm (the upper limit was not easily tested.) It should also be noted that the position data was more accurate when using the HLDx-2 type landmark at heights of over 3300mm.

Field of View (FOV) & Precision Test: This test was designed to determine the size of the field of view, or in other words, the localization range per landmark. Table I shows the limits for several heights. The first thing to notice is that the camera was not heading what we consider the center of reference because positive and negative values are different.

Figure 9 clearly demonstrates an important property of the system near the field of view limits; the precision is considerably reduced within the final 20 – 30cm (for a height

TABLE I
FIELD OF VIEW AT DIFFERENT HEIGHTS (UNITS ARE IN CM AND DEG)

Heights	X_{min}	X_{max}	Y_{min}	Y_{max}	X_{FOV}	Y_{FOV}
250m	-147	134	-167	154	58,65	62,04
300m	-177	160	-170	149	58,61	55,95
350m	-207	--	-206	180	--	57,70

X_{min} and X_{max} should be the same but with different sign however, the difference is because the camera was not focusing what we consider the center (0,0). The opening angle for the field of view in both axis is similar and close to 60 degrees.

of 2.5m) of the field of view edge. At the limit the size of the range of measured y-values is approximately 13cm.

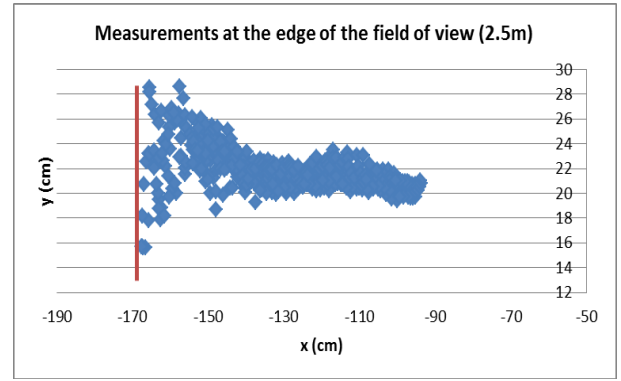


Fig. 9. Precision of measurements is clearly reduced at the edge of the field of view.

Accuracy test: This test was designed to determine how accurate the StarGazer system is by measuring a change in the tag position in both axes. Table II lists the results obtained when moving the landmark $\Delta x = 20cm$ at different positions in the field of view at four different heights. Each Δx is averaged over 50 individual measurements. Landmark type HLDx-1 was used for the 2.5m and 3.0m tests, while type HLDx-2 was used for the 3.5m and 4.5m tests.

We conservatively estimate that the maximum error in position was measured by hand is approximately 0.5mm.

The results demonstrate the high level of accuracy that the StarGazer system can achieve. For heights up to 3.5m the average measurement error was less than 1%. This translates to under 2mm, in most cases, over a 20cm change in position.

At the larger height of 4.5m the error increased significantly to about 6%, or about 1.1cm over the 20cm change in position.

To gain an idea of the worst case error in the StarGazer measurements, the largest error over all the measurements taken in each test was also calculated. In all cases but one, the largest error was between 1 & 3 times the average error. The largest error was still under 2% for heights up to 3.5m. In the 4.5m test case the largest error was only very slightly greater than the average error. This could be due to the different landmark type.

Repeatability test: This test was designed to determine the repeatability or precision of the device. By this we mean

TABLE II

FIELD OF VIEW AT DIFFERENT HEIGHTS (UNITS ARE IN CM AND DEG)

Height	Threshold Value	Avg. Δx measured	Avg. % error	Largest % error
2.5m	225	20.06	0.30	0.80
	235	20.03	0.16	0.30
	245	20.01	0.05	0.30
3.0m	225	20.11	0.54	1.40
	235	19.93	0.36	0.85
	245	19.79	1.03	1.45
3.5m	225	20.12	0.59	1.10
	235	20.18	0.89	1.25
	245	20.21	1.04	1.55
4.5m	225	21.11	5.55	6.65
	235	21.14	5.72	5.85
	245	21.17	5.85	5.85

The Threshold Value is a number between 0 & 255 and it corresponds to the level for rejecting external interference in the images obtained by the device. The user manual recommends a value in the range 210 – 240 depending on the application.

the ability to obtain the same measurement in the exact same position. This can be quantified by the spread of a number of measurements. In our case we look at the standard deviation and range of values. The repeatability was tested at various positions in the field of view of the device to determine if it is related to the distance from the center of the field of view to the landmark. The larger standard deviations are observed at the positions furthest from the center, and nearing the edge of the field of view. Another important thing to notice is that the standard deviations of the y-position measurements are much larger than those of the x-position measurements when we move the tag along X-axis.

Figure 10 presents the variation in the standard deviation of the measurements at each x-position averaged over different threshold values. Even though this is not the behavior for all thresholds it works for the averaged values.

B. The WLS system

Details about this low cost indoor localization system can be seen in [13]. In this section we contrast the results with the ones presented here for the StarGazer.

Height. For the range tested (2m to 4m.) both systems work fine. However, landmarks for the WLS need to be bigger because of the minimum tag distance required for the WiiMote to detect two tags instead of integrate them in one tag. Table III shows the minimum tag distance required for different height values.

Field of view. The WiiMote has a field of view of 45 degrees, while the StarGazer was found to have a field of view of approximately 60 degrees. This means that fewer landmarks are needed at the same ceiling height with the StarGazer. It would not be a bad idea, however, to only use 45 degrees of the StarGazer's field of view because as we have seen, the precision deteriorates at the limits of the FOV.

TABLE III

MINIMUM DISTANCE BETWEEN TAGS (UNITS ARE IN CM AND DEG)

Height	Minimum distance	Angle
50	1	1, 1458774
150	2, 9	1, 1076839
200	4, 4	1, 26045631
250	4, 7	1, 07712893
300	5, 3	1, 01219911
350	4, 7	0, 76938891
400	5, 7	0, 81645104
450	5, 7	0, 72573684
500	6, 2	0, 71045856

The distance depends on the height.

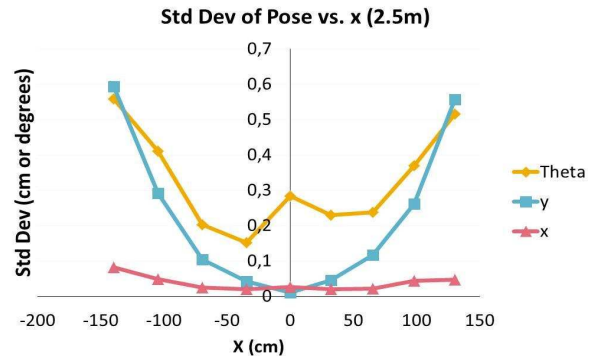


Fig. 10. Average standard deviation along the x-axis.

Refresh rate. The WiiMote image refresh rate is $100Hz$, while the StarGazer reports localization data at $10 - 20Hz$. The important difference is that the WiiMote image data must be processed, while the StarGazer data is already in usable form. Both systems are sufficiently fast for robots that move at under $1m/s$.

Power efficiency is also a very important issue since the illuminator array is installed on-board the robot. The stargazer in general needs less power ($0.84 \sim 1.5W$) than the WLS.

In the WLS we have information about the position of each tag that composes the landmark. This can be used to implement other localization systems such as the ones presented in [13] However, special care has to be taken when other IR sources can be found in the FOV such as sunlight when the robot navigates close to windows or shiny metal bars on the ceiling that can reflect the IR led light. These can produce false tag readings when the robot is below them. However, it is very unlikely that a pattern resembling a valid landmark is produced.

In general, we can conclude that both solutions work quite well in most of the environments where the different tests have been performed. Since the StarGazer is designed specifically for localization, it outperforms the WLS in most of the desired localization characteristics. The WiiMote also contains a $\pm 3g$, 8-bit 3-axis accelerometer operating at $100Hz$ and an expansion port for even more capability that could be integrated for robot localization even though it is not included in this paper.

VI. THE WLS SYSTEM WITH RANDOM TAGS

One of the advantages of using the WiiMote is that the position of the four IR light sources is provided. Therefore, unlike the landmarks in figure 2, the retro-reflective tags can be arranged in any other way. For example, if the distance between the tags is increased, a more accurate robot angular position can be obtained. That will solve one of the main problems of the localization system described above. Tags can even be randomly attached to the ceiling without being part of a landmark. In this case, an issue to consider is the best tag distribution bearing in mind the following goals:

- The robot should never “see” more than four tags.
- To completely locate itself, the robot should detect at least two tags. Therefore, an ideal situation will be if the robot sees always at least two tags.

A. TAG distribution

There are some tag distributions for some areas such as corridors that can easily meet both conditions because of the particularly geometry of the area. However, finding the distribution that meets both conditions in big open areas is a bigger challenge.

For localization, tags are distributed based on a pattern such that the location of each tag can be easily computed [22]. Here, we use and analyze the distribution pattern shown in figure 11.

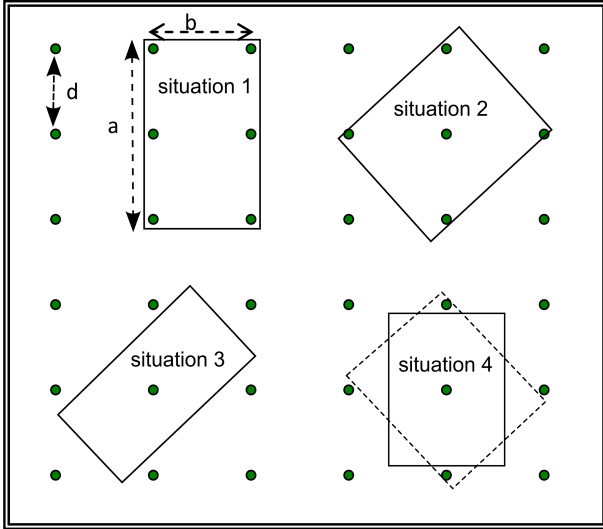


Fig. 11. Tag distribution for an opening area.

The tag distribution of figure 11 is defined by the distance between tags (d). To obtain the possible values of this distance we should use the following two theorems:

Theorem 1: Considering that the area seen by the WiiMote is a rectangle of sides a (1024) and b (768). If the distance d satisfies the following equations:

$$d > \frac{b}{\sqrt{2}} = 544 \quad (3)$$

$$d < b = 768 \quad (4)$$

Then:

- 1) The WiiMote field of view always includes at least one tag.
- 2) The WiiMote field of view never includes more than four tags.

Proof of the second point can be established constructing the combinations of all possible five neighbour tags. None of them fit in the rectangle of sides b and a . The limit situation where restriction (3) is not satisfied is presented in figure 11 (situation 2). Therefore, with restriction (3), the robot will never see more than four tags. Proof of the first point can be obtained because there is not such an area without tags if restriction (4) holds.

Another situation to avoid is the robot turning in the same place incrementing the odometry angular error. That could happen if the robot rotates on one point with a tag right above it.

Theorem 2: The robot cannot turn more than 90 degrees without detecting two tags (situation 4 dotted rectangle) if restriction (5) holds.

$$d < \frac{1}{2}\sqrt{a^2 + b^2} = 640 \quad (5)$$

Proof of this theorem can be easily obtained from situation 4 of figure 11. If the robot is rotating right under the tag and restriction 5 holds, it will not rotate more than 90 degrees without detecting more than two tags.

Another characteristic of this distribution is that the robot cannot move linearly more than $(2d\sqrt{2}) - a$ while detecting only one tag. This corresponds to the situation of rectangle 3 in figure 11.

B. The Localization system

A possible deterministic localization algorithm considers two different situations depending on the number of tags detected:

- One tag: only the robot’s planar (x,y) coordinates are corrected in order to fit the tag position.
- More than one tag: the robot pose is obtained from two tag readings.

This deterministic solution considers only two tag readings. Other possible solutions that take into account all tag readings have also been considered. One example is shown in fig 12. First, the robot is moved from its last position according to the odometry information. Next, the camera coordinates of the map tags in the surrounding of the robot are obtained. In figure 12 these tags (A,B,C) are represented as ‘*’ and the observed tags (a,b,c) as ‘o’. Let’s assume the tag reading points are part of a rigid solid, the localization problem can be then seen as finding the position of this “solid” that minimizes the sum of distances d_{Aa} , d_{Bb} and d_{Cc} . This is, knowing that (A,B,C) are constants, we have to minimize:

$$f(a_x, a_y, b_x, b_y, c_x, c_y) = d_{Aa} + d_{Bb} + d_{Cc} \quad (6)$$

where

$$\begin{aligned} d_{Aa} &= (A_x - a_x)^2 + (A_y - a_y)^2 \\ d_{Bb} &= (B_x - b_x)^2 + (B_y - b_y)^2 \\ d_{Cc} &= (C_x - c_x)^2 + (C_y - c_y)^2 \end{aligned} \quad (7)$$

Under the solid rigid distance restrictions:

$$\begin{aligned} d_{ab} &= (a_x - b_x)^2 + (a_y - b_y)^2 = \text{const} \\ d_{ac} &= (a_x - c_x)^2 + (a_y - c_y)^2 = \text{const} \\ d_{bc} &= (b_x - c_x)^2 + (b_y - c_y)^2 = \text{const} \end{aligned} \quad (8)$$

A Newton-Raphson method has been implemented that converges quite fast since the reading points are very close to the tag maps. However, it is still quite a CPU intensive process because it includes inverting an 11x11 matrix in each step.

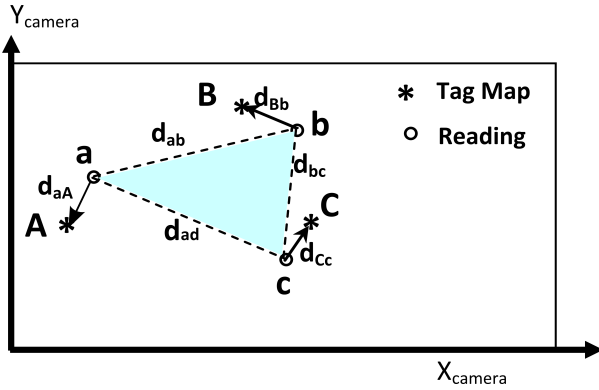


Fig. 12. Localization problem seen as the equilibrium point of a solid defined by the readings and anchored to the Tags in the map by springs. A,B and C are the tag maps and a,b,c are the readings.

Another way to view this is as a solid anchored with springs that join the readings to the map tags. In this case, the localization problem is equivalent to find the equilibrium position of the solid. For example, in the case of detecting three tags like in figure 12, the position could be obtained solving the following equation system under the restrictions of equation (8).

$$\sum F_x = (A_x - a_x) + (B_x - b_x) + (C_x - c_x) = 0 \quad (9)$$

$$\sum F_y = (A_y - a_y) + (B_y - b_y) + (C_y - c_y) = 0 \quad (10)$$

$$\begin{aligned} \sum M &= (A_x - a_x)a_y + (A_y - a_y)a_x + (B_x - b_x)b_y - \\ &(B_y - b_y)b_x + (C_x - c_x)c_y - (C_y - c_y)c_x = 0 \end{aligned} \quad (11)$$

The first two equations (9), (10) correspond to the sum of forces and the third (11) is the sum of momentum. However, finding the analytical solution of this equation system is not an easy task and it has several solutions because the solid has several equilibrium points. Therefore, the sum of distances d_{Aa} , d_{Bb} and d_{Cc} has to be obtained for each possible solution and then the one with the minimum value chosen. This task becomes more difficult when the WiiMote detects four tags.

The deterministic solution considering only two tag readings works very well for environments where no other IR sources can be found. If the building has a flat floor and a low-jerking robot is used, an accurate position is obtained.

VII. CONCLUSIONS

Two infrared landmark measurement systems (the Hagonic StarGazer and the WiiMote based system,) have been evaluated. The landmarks for both systems include a set of retroreflective tags with the same pattern. Comparing both systems we can conclude the following points:

- The WLS landmarks are limited to four tags. This limits the number of different landmark ID's to 5 for this case. The StarGazer system can use 25=32 IDs.
- The StarGazer system provides two kinds of landmarks: Landmark type HLDx-1 was usable for heights over the range of 2200mm to 4300mm and HLDx-2 worked from 3300mm. The WiiMote landmarks need to be bigger for the same distance (high) than the StarGazer landmarks.
- The StarGazer system has a wider FOV (about 60°) than the WLS (about 45°) even though the precision deteriorates at the limits of the FOV.
- The StarGazer system has a higher level of accuracy for the landmarks considered here. For heights of up to 3.5m the average measurement error was less than 1%. However, for the WLS system we can build landmarks with a bigger distance between tags obtaining better orientation accuracy.
- The WiiMote image refresh rate is 100Hz, while the StarGazer reports localization data at 10-20Hz.

In the WLS we have information about the position of each tag that composes the landmark. This can be used to implement other localization systems such as the ones presented in [13]. However, special care has to be taken when other IR sources can be found in the FOV such as sunlight when the robot navigates close to windows or shiny metal bars on the ceiling that can reflect the IR led light. These can produce false tag readings when the robot is below them. However, it is very unlikely that a pattern resembling a valid landmark would be produced.

In general, we can conclude that both solutions work quite well in most of the environments where the different tests have been performed. Since the StarGazer is designed specifically for localization, it outperforms the WLS in most of the desired localization characteristics. The WiiMote also contains a +/-3g 8-bit 3-axis accelerometer operating at 100Hz and an expansion port for even more capability that could be integrated for robot localization even though it is not included in this paper.

Finally, we have also proposed a different tag arrangement that was used with the WLS. A drawback of a localization system based only on the reflective landmarks is the number of tags needed. For a typical ceiling, four meters high and a robot half a meter tall, the field of view of the IR camera is an area of approximately 9 square meters. This means that, depending on the robot working area, a large number of tags might be needed.

The WLS system with random tags proposed does not deal well with noisy readings. A Kalman or a particle filter that is able to deal with noisy readings, can take into account information of all detected tags including odometry and allows for a reduction in the number of tags needed. A possible

solution for this case is presented in [13].

ACKNOWLEDGMENT

We would like to thank all the people that have influenced this work. This work has been partially founded by Ministerio Español de Ciencia y Tecnología under proyect DPI2010-20466-C02-01.

REFERENCES

- [1] S. Thrun, W. Burgard, D. Fox, "Probabilistic Robotics", Ed. MIT Press, Cambridge, Massachusetts, 2006.
- [2] B. Hofmann-Wellenhof, K. Legat and M. Wieser, "Navigation -Principles of positioning and guidance-", Springer-Verlag, Wien, 2003.
- [3] Y. Zhang, Y. Luo, and J.F. Wang, A novel method of range measuring for a mobile robot based on multi-sensor information fusion. *Journal of Physics: Conference Series* 13, pp.127-132. ISSN: 1742-6588, 1742-6596 (Online).
- [4] S. Thrun, W. Burgard, D. Fox, "Monte Carlo localization with mixture proposal distribution." In proceedings of the AAAI National Conference on Artificial Intelligence, Austin, TX. AAAI.
- [5] I.J. Cox. Blanche, "An experiment in guidance and navigation of an autonomous robot vehicle". *IEEE Transactions on Robotics and Automation*, vol.7,NO.2 ,pp.193–204, 1991
- [6] G. Kitagawa, "Monte carlo filter and smoother for non-gaussian nonlinear state space models," *Journal of Computational and Graphical Statistics* 5(1), pp. 1–25, 1996.
- [7] J. López, C. Watkins, D. Pérez and M. Díaz-Cacho Medina. "Implementing and evaluating an IR landmark positioning system in the RIDE architecture". *Proceedings of the XIII Workshop en Agentes Físicos*, (WAF 2012). pp:23-28.
- [8] J. López, D. Perez (2011) and E. Zalama, "A framework for building mobile single and multi-robot applications", *Robotics and Autonomous Systems* 59, 151-162, (2011).
- [9] H. Moravec and A. Elfes, "High resolution maps from wide angle sonar," in *Proceedings of the 1985 IEEE International Conference on Robotics and Automation*, 1985.
- [10] S. Thrun, "Learning occupancy grids with forward models," in *Proceedings of the Conference on Intelligent Robots and Systems (IROS'2001)*, 2001.
- [11] Johnny Chung Lee, "Hacking the Nintendo Wii Remote," *IEEE Pervasive Computing*, vol. 7, no. 3, pp. 39-45, July-Sept. 2008, doi:10.1109/MPRV.2008.53
- [12] B. Tas, N. Altıparmak, A.S. Tosun, "Low cost indoor location management system using infrared leds and Wii Remote Controller", *Performance Computing and Communications Conference (IPCCC)*, 2009 *IEEE 28th International* , vol., no., pp.280-288, 14-16 Dec. 2009.
- [13] Joaquín López, Diego Pérez, Eduardo Zalama and Jaime Gómez-García-Bermejo. "Low Cost Indoor Mobile Robot Localization System". *Proceedings of The 2011 11th International Conference on Intelligent Systems Design and Applications (ISDA 2011)* . pp: 1134-1139.
- [14] R. Bonasso, D. Kortenkamp, D. Miller and M. Slack, "Experiences with an architecture for intelligent, reactive agents". *Journal of Artificial Intelligence Research* Vol 9(1), pp: 237-256, 1997
- [15] M. Montemerlo, N. Roy, S. Thrun, *Perspectives on standardization in Mobile Robot Programming : The Carnegie Mellon Navigation (CARMEN) Toolkit*. *Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems, (IROS 2003)*. 27-31 Oct. 2003. Vol. 3, 2436- 2441
- [16] A.R. Diéguez, J.L. Fernández, R. Sanz, "A global motion planner that learns from experience for autonomous mobile robots", *Robotics & CIM* Vol. 23 pp: 544–552 (2007), Ed. Elsevier.
- [17] J.L. Fernández, R. Sanz, J.A. Benayas and A.R. Diéguez, "Improving collision avoidance for mobile robots in partially known environments: the beam curvature method." *Robotics and Autonomous Systems*. vol. 46, pp. 205-219, April 2004.
- [18] J. L. Fernández, Maria J. Souto, Diego P. Losada, R. Sanz, E. Paz. "Communication framework for sensor-actuator data in mobile robots". *Proceedings of the 2007 IEEE International Symposium on Industrial Electronics*, pp:1502-150. ISBN: 1-4244-0755-9.
- [19] Joaquín L. Fernández, Rafael S. Dominguez, Enrique Paz Domonte and Carlos Alonso. "Using hierarchical binary Petri nets to build robust mobile robot applications: RoboGraph". *Proceedings of ICRA 2008* pp:1372-1377.
- [20] S. Thrun, D. Fox, W. Burgard, and F. Dellaert. *Robust Monte Carlo localization for mobile robots*. *Artificial Intelligence*, 101:99–141, 2000.
- [21] <http://wiibrew.org/wiki/Wiimote/Library> Accessed March 31, 2012.
- [22] Nejad, K.K.; Xiaohong Jiang; Kameyama, M.; , "Non-blocking tag scanning for passive RFID localization," *Intelligent Systems Design and Applications (ISDA)*, 2011 11th International Conference on , vol., no., pp.1140-1145, 22-24 Nov. 2011.