



Contents lists available at SciVerse ScienceDirect

## Expert Systems with Applications

journal homepage: [www.elsevier.com/locate/eswa](http://www.elsevier.com/locate/eswa)

## Mobile robot map building from time-of-flight camera

Sergio Almansa-Valverde<sup>a</sup>, José Carlos Castillo<sup>a</sup>, Antonio Fernández-Caballero<sup>a,b,\*</sup>

<sup>a</sup>Instituto de Investigación en Informática de Albacete (I3A), Universidad de Castilla–La Mancha, 02071 Albacete, Spain

<sup>b</sup>Departamento de Sistemas Informáticos, Escuela de Ingenieros Industriales de Albacete, Universidad de Castilla–La Mancha, 02071 Albacete, Spain

### ARTICLE INFO

#### Keywords:

Mapping  
Mobile robot  
Time-of-flight (ToF) camera  
Occupancy grid

### ABSTRACT

A map building algorithm for mobile robots is introduced in this paper. The perceived environment is represented in a map containing in each cell a probability of presence of an object or part of an object. The environment is represented as a collection of modular occupancy grids which are added to the map as far as the mobile robot finds objects outside the existing grids. In this approach a time-of-flight (ToF) camera is exploited as a range sensor for mapping. Indeed, one of the areas where ToF sensors are adequate is in obstacle avoidance, because the detection region is not only horizontal but also vertical, allowing to detect obstacles with complex shapes. The main steps of the map building algorithm are extensively described in the paper. The results of testing the algorithm are considered in two different indoor environments.

© 2012 Elsevier Ltd. All rights reserved.

### 1. Introduction

The fundamental capacity a mobile robot must possess while navigating in an unknown environment is self-localization as well as maintaining a notion of its surrounding environment (Begum, Mann, & Gosine, 2008; Gascueña & Fernández-Caballero, 2011). Mapping techniques for mobile robots can be roughly classified according to the map representation and the underlying estimation technique (Wurm, Stachniss, & Grisetti, 2010). One popular map representation is the occupancy grid. Whereas such grid-based approaches are computationally expensive and typically require a huge amount of memory, they are able to represent arbitrary objects. Feature-based representations are attractive because of their compactness. This is a clear advantage in terms of memory consumption and processing speed. However, such systems rely on predefined feature extractors, which assume that some structures in the environments are known in advance.

There are many publications on the use of vision for mapping and localization in mobile robotics. The so-called visual SLAM (Chatterjee & Matsuno, 2010; Chatterjee, Ray, Chatterjee, & Rakshit, 2011; Kaess & Dellaert, 2010) uses monocular (Davison, Reid, Molton, & Stasse, 2007; Eade & Drummond, 2009; Sim, Elinas, & Little, 2007; Sim & Little, 2009; Williams et al., 2009), stereo-based (Fernández-Caballero, López, & Saiz-Valverde, 2008; López-Valles, Fernández, & Fernández-Caballero, 2007; Marks, Howard, Bajracharya, Cottrell, & Matthies, 2007; Moreno, Blanco, & Gonzalez, 2009;

Paz, Piniés, Tardós, & Neira, 2008; Schleicher, Bergasa, Ocaña, Barea, & López, 2010) and omnidirectional sensors (Feng, Chen, & Hornig, 2010; Goedemé, Nuttin, Tuytelaars, & Van Gool, 2007). One possible monocular solution is the use of a time-of-flight (ToF) range camera. Indeed, ToF camera gives depth information per pixel which make them ideal for background foreground segmentation, as in general the depth defines the subject from background in a much more basic way than the light intensity does (Guethmundsson et al., 2010). Intensity images are on the other hand affected by colors, lighting, reflections and shadows in almost every normal scenario (Fernández-Caballero, Castillo, Serrano-Cuerda, & Maldonado-Bascón, 2011; López, Fernández-Caballero, Fernández, Mira, & Delgado, 2007). Therefore, ultrasonic sensors are widely used in robotics to avoid collisions and for map building purposes (Benet, Martínez, Blanes, Pérez, & Simó, 2005).

A very recent technical report reviews the state-of-the-art in the field of ToF cameras (Foix, Alenyá, & Torras, 2010). A usual framework is to install the camera on a mobile robot and use it for robot navigation and mapping. One of the areas where ToF sensors are adequate is in obstacle avoidance, because the detection region is not only horizontal but also vertical, allowing to detect obstacles with complex shapes. Some applications also benefit from the metric information obtained with depth images. Thanks to the larger vertical field of view of ToF cameras, difficult obstacles (like tables) are better detected by a ToF camera than by a 2D laser scanner (Weingarten, Gruener, & Siegart, 2004). ToF sensors have been used successfully as the unique sensor in some mobile robotic applications, despite their characteristic limited resolution. For mapping purposes, ToF sensors are very interesting because they allow to extract geometric features. Most of the applications extract planar regions using both intensity and depth images. In

\* Corresponding author at: Instituto de Investigación en Informática de Albacete (I3A), Universidad de Castilla–La Mancha, 02071 Albacete, Spain. Tel.: +34 967599200; fax: +34 967 599224.

E-mail address: [antonio.fdez@uclm.es](mailto:antonio.fdez@uclm.es) (A. Fernández-Caballero).

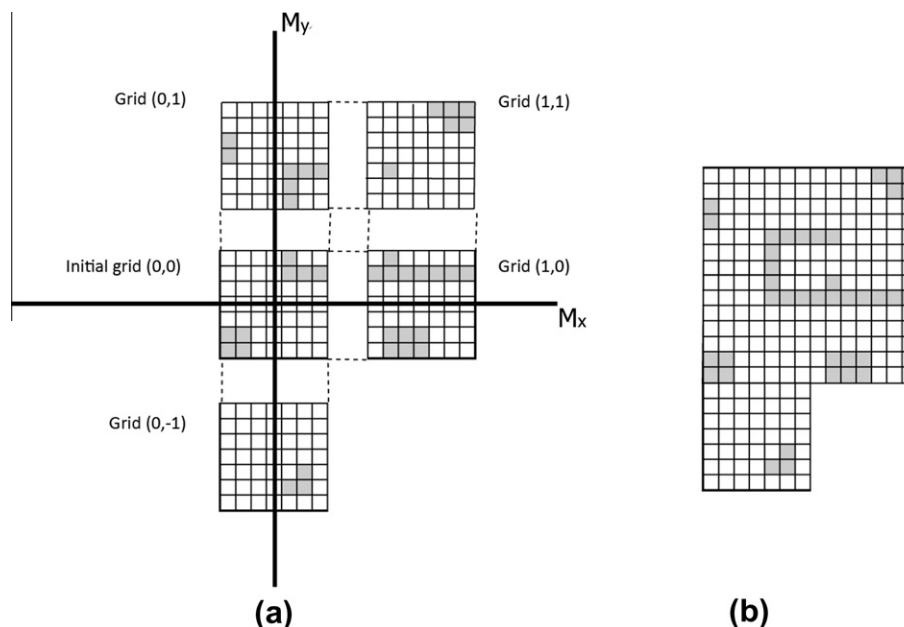


Fig. 1. Environment model. (a) Addition of new grids to the map. (b) Resulting environment map.

a few papers (May, Droschel, Holz, Wiesen, & Fuchs, 2008; May et al., 2009) different methods are explored to improve pose estimation. The normal of the extracted planes is also used (Hedge & Ye, 2009) to detect badly conditioned plane detection, as horizontal planes in a staircase. Also with potential applications to SLAM, a corner filtering scheme combining both the intensity and depth image of a ToF camera has been proposed (Gemeiner, Jojic, & Vincze, 2009).

Our proposal deals with the specific characteristics of ToF cameras (Lange, 2000) to solve the mapping problem. Consequently, it has been necessary to study the performance of this kind of sensor and its behavior in realistic environments (detection range, tolerance, noise measure, among others). The rest of the paper is organized as follows: Section 2 introduces a general view of the mapping proposal. Then, Section 3 presents the mapping algorithm in detail. After this, Section 4 offers some promising results obtained so far. Lastly, in Section 5 some conclusions are offered.

## 2. Mapping from ToF camera

The ToF camera is exploited as a range sensor for mapping. The ToF camera used as a range sensor provides a powerful tool for detecting objects in front of the robot by measuring the distance towards them. Indeed, the ToF camera is considered as a range sensor due to the simplicity of translating the  $(x,y,z)$  coordinates of each image pixel provided by the camera into map coordinates. The perceived environment is represented in a map containing in each cell a probability of presence of an object or part of an object. In this sense, an occupancy grid model has been used to represent the environment. This model is selected as it represents an indoor-oriented model and it is easy to implement. Occupancy grid models present two problems. First, the discretization error has been minimized by reducing the size of the cells; it is practically irrelevant for the navigation task. And second, the memory requirements have been mitigated by creating the grids only as the observed environment grows.

As the size of the environment is unknown a priori, it is not possible to create a fixed-size occupancy grid. The implemented solution represents the environment as a collection of modular occupancy grids which are added to the map as far as the robot

finds objects outside the existing grids. Therefore, when the robot starts the exploration there exists just one grid, and all necessary grids are added according to the size of the explored environment. Each occupancy grid has the same size and number of cells and is placed in a specific area given by its global coordinates  $(M_x, M_y)$  in the environmental representation. So every point in the environment is located in one and only one grid. A representation of this system is shown in Fig. 1.

For each grid, every cell contains an occupancy value which represents the probability that the corresponding area is occupied by an obstacle, where a value of 0.00 indicates the absolute certainty that the cell is free and a value of 1.00 the certainty that the cell is occupied. Although the parameters are configurable, the experiments are carried out with  $10 \times 10$  cm cells and 101 cells per grid. The number 101 of cells in each grid guarantees symmetric grids, assuming that each grid has a local coordinate system with origin in the center of the grid. That way the central column/row corresponds to coordinates  $(x=0, y=0)$  and each quadrant contains  $50 \times 50$  cells.

## 3. The map building algorithm

In the previous section, the environment model was introduced. The current section is devoted to the description of the mapping algorithm. Fig. 2 shows a flow diagram of the algorithm. The main steps are explained next.

### 3.1. Selection of the pixels distance information

As aforesaid, the ToF camera is used as a range sensor as input to the mapping algorithm. The ToF camera provides as output a matrix where each position represents the three-dimensional coordinates  $(x,y,z)$  (in meters) of a system. Here the camera is the origin of coordinates,  $x$  varies along the horizontal axis,  $y$  varies along the vertical axis and  $z$  is the distance from the plane defined by the  $x$  and  $y$  axes.

From the camera point of view there is distance information of points at different heights along the  $y$  axis. Thus, it is possible to create different maps depending on the height interval of the detected data to be mapped into the occupancy grid. This is done

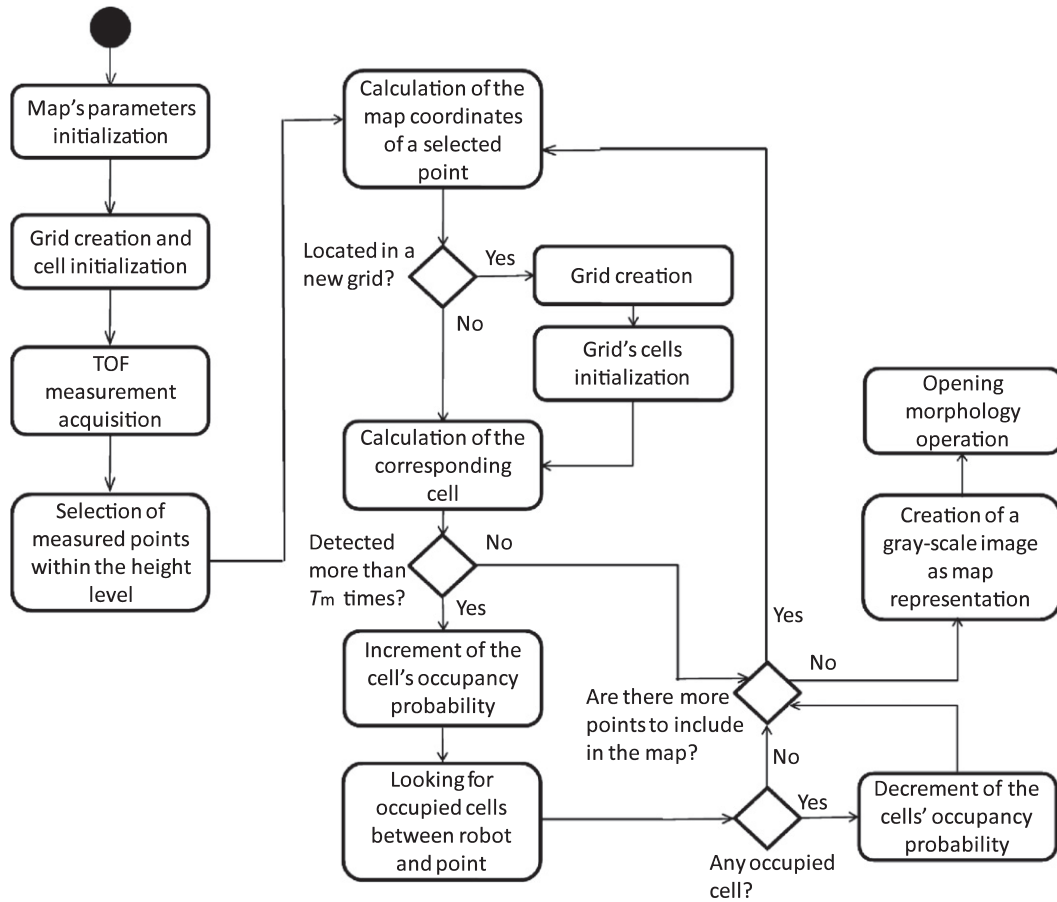


Fig. 2. Flow diagram of the mapping algorithm.

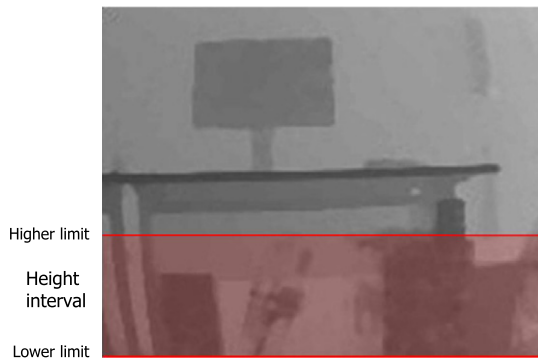


Fig. 3. Height interval on a distance image.

by comparing the value of the  $y$  coordinate of each pixel with the desired height interval. The height interval is chosen according to the purpose of the map. For instance, if the height interval is adjusted to the robot's height, the map shows the real obstacles in the robot's path, while objects above its height will not be shown. If navigating with a robot smaller than a table, the map would only show as occupied the space of the table's legs but not the board; so the robot is able to move under the table as shown in Fig. 3, where only the measures within the interval (in red color<sup>1</sup>) are represented in the map. This would be impossible if the whole table was represented in the occupancy grid. An alternative is to simul-

taneously create several maps with different height intervals to achieve a more complete and versatile representation of the environment.

### 3.2. Calculation of the map coordinates

After selecting the pixels' height range to create the map, their positions in the map are calculated. Let us assume that the robot position  $(m_r, x_r, y_r, \theta)$  is known, where  $m_r$  is the occupancy grid where the robot is located,  $x_r$  and  $y_r$  represent the position within that grid, and  $\theta$  is the robot orientation with respect to the  $y$  axis in the grid. On the other hand, the ToF camera provides the coordinates of point  $(x_{ToF}, y_{ToF}, z_{ToF})$  as its position relative to the robot's localization. From these two groups of coordinates, it is possible to calculate the position of a point  $p(m, x, y, h)$  in the map, being  $m$  the occupancy grid where the point is located,  $x$  and  $y$  represent the positions within that grid and  $h$  is the height of the point.

Fig. 4 shows a visual representation of the parameters calculated, and the correspondence between ToF coordinates and map coordinates. In Fig. 4a the relation between the robot and the observed point, dependent on ToF coordinates, is presented. Fig. 4b shows the same point located in its corresponding position in the map (which consists of one or more grids).

In order to represent each point detected by the camera some calculations are performed. First, the distance  $D$  between the robot and the point (projected to the ground plane) is calculated as follows:

$$D = \sqrt{x_{ToF}^2 + z_{ToF}^2} \quad (1)$$

<sup>1</sup> For interpretation of the references to color in this figure, the reader is referred to the web version of this article.

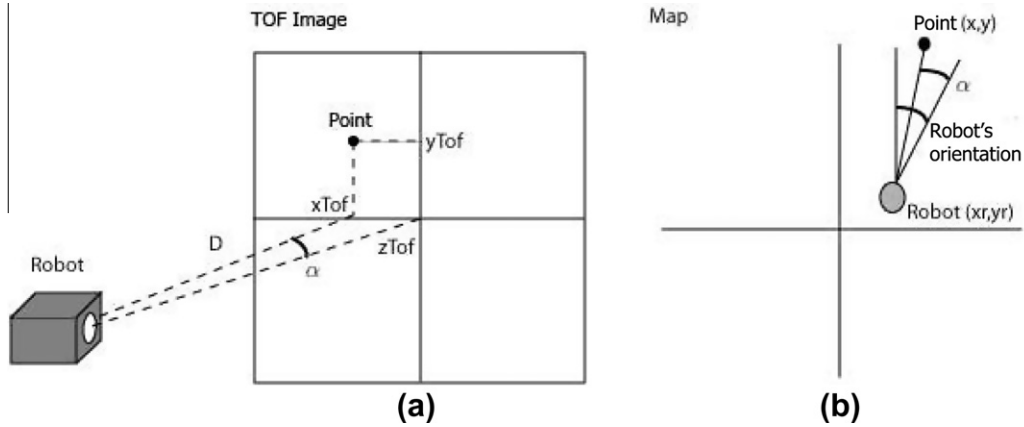


Fig. 4. Correspondence of ToF and map coordinates of a random observed point. (a) Point  $p$  represented by its ToF coordinates. (b) Point  $p$  represented by its map coordinates.

Once the distance is known, the angle  $\alpha$  between the robot orientation and the segment given by  $D$  is obtained:

$$\sin(\alpha) = \frac{x_{ToF}}{D} \rightarrow \alpha = \arcsin\left(\frac{x_{ToF}}{D}\right) \quad (2)$$

And, finally, through the following equations, the position of the point in the map is also calculated:

$$\begin{aligned} x &= x_r + D \sin(\theta + \alpha) \\ y &= y_r + D \cos(\theta + \alpha) \\ h &= y_{ToF} \end{aligned} \quad (3)$$

Next, it is necessary to know whether each new point is located on an existing grid. Otherwise, it is necessary to create a new one. Assuming that the grid  $m$  where the robot is located has global coordinates  $(m_x, m_y)$ ,  $T$  is the length of the grid's side expressed in meters and  $m'$  is the grid that contains the found point with global coordinates  $(m'_x, m'_y)$ , then:

$$m'(m'_x, m'_y) = \begin{cases} x = x - T, & m'_x = m_x + 1, & m'_y = m_y, & \text{if } x_r + x > T/2 \\ x = x + T, & m'_x = m_x - 1, & m'_y = m_y, & \text{if } x_r + x < -T/2 \\ y = y - T, & m'_x = m_x, & m'_y = m_y + 1, & \text{if } y_r + y > T/2 \\ y = y + T, & m'_x = m_x, & m'_y = m_y - 1, & \text{if } y_r + y < -T/2 \\ m'_x = m_x, & m'_y = m_y, & & \text{otherwise} \end{cases} \quad (4)$$

More than one of the previous conditions could be fulfilled simultaneously when the detected point is more than one grid away or in a contiguous grid in the diagonal. For example, if the situation is as shown in Fig. 5 with the robot in the initial position and  $T = 7$ , then  $x_r + x = 5 > 3.5$  and  $y_r + y = 8 > 3.5$ . Therefore, the first and third equations are applied resulting on  $(m'_x = 1, m'_y = 1)$  as the global coordinates of the grid where the point is located, and  $x = x - T = -2$  and  $y = y - T = 1$  as local coordinates of the cell which contains the point.

### 3.3. Update of the occupancy probability of the grid cells

Once the coordinates of every point have been calculated, the new observations must be represented in the grid, updating the occupancy probability of the affected cells. As each cell covers several squared centimeters of the map, it is usual that more than one detected point is located into the same cell for the same observation. Consequently, the occupancy probability of each cell is increased as many times as points are detected to belong to it.

As aforesaid, the occupancy probability of a cell ranges from 0.00 to 1.00, where the greater values indicate higher likelihood of being occupied. Every time a point is detected within a cell,

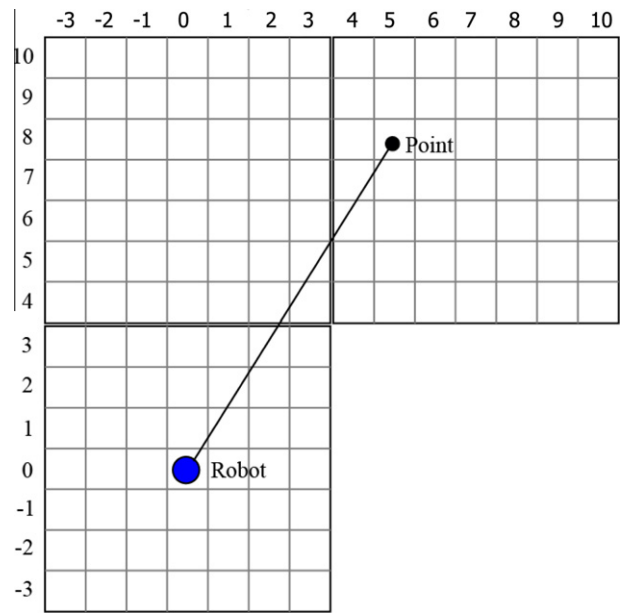
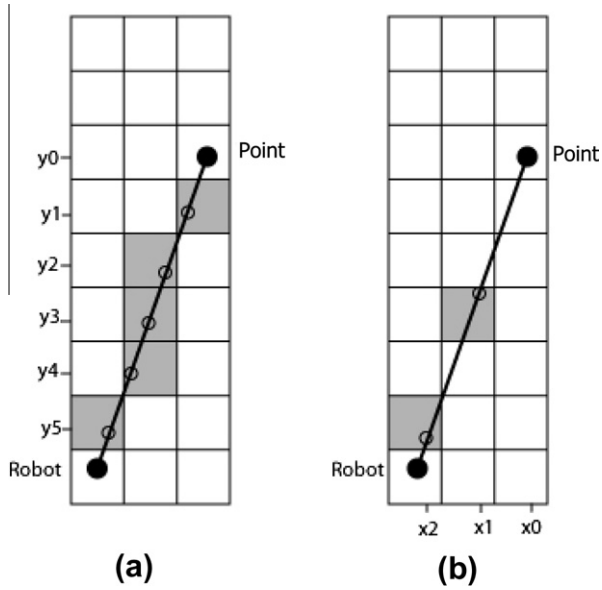


Fig. 5. Detection of a point in a different grid.

the probability of that cell is increased in  $\tau_i$ . It is recommended keeping  $\tau_i$  low (ranging from 0.10 to 0.25) because several points corresponding to the same cell are generally found in the same observation of the ToF camera. So the probability of the cell is increased several times, achieving a high occupancy probability without the need of a higher  $\tau_i$ . Indeed, a higher value would reach the highest priority with a few observations. There is another reason for keeping the value low. If the camera measurement is noisy and the occupancy probability of a cell is erroneously increased, with a low  $\tau_i$  the error will be small as these points are usually scattered. The algorithm also implements a threshold  $\tau_m$  to avoid a wrong representation of the environment due to noisy data. This global threshold  $\tau_m$  (independent to cells values), indicates the minimum number of points that must be detected within a cell to let the occupancy probability increase. Therefore, given two consecutive observations, the occupancy probability for a given cell  $(m, x, y)$  at time instant  $t$ , that is  $P_t(m, x, y)$ , with  $N$  points belonging to the cell is calculated with Eq. 5.

$$P_t(m, x, y) = \begin{cases} P_{t-1}(m, x, y) + \tau_i, & \text{if } N \geq \tau_m \\ P_{t-1}(m, x, y), & \text{otherwise} \end{cases} \quad (5)$$



**Fig. 6.** Line segment between the robot and the detected point with an angle between  $45^\circ$  and  $135^\circ$ . (a) Cells selected decrementing on  $y$  axis. (b) Cells selected decrementing on  $x$  axis.

Decreasing the occupancy probability of grid cells is also necessary. This provides some robustness against dynamic objects in the scene and noise in ToF camera measurements that might lead to erroneous map data. The general idea is that a cell is not occupied when the robot gets information from behind this cell. To find out whether an erroneously occupied cell is free (because the measurement is noisy, or because a dynamic object is detected but has changed its position), the equation of the straight line between a detected point and the robot is calculated. Then, the value of one of the coordinates  $x$  or  $y$  is changed by intervals equivalent to the cell size, and the other coordinate is calculated using the equation of the line:

$$y = m \cdot x + b \quad (6)$$

From those coordinates, the correspondent cells in the segment between the robot position and the point are calculated, and their occupancy probabilities are decreased by  $\tau_d < \tau_i$ . It is worth mentioning that it is necessary to first calculate  $x$  from the line's equation and from the different values of  $y$ , and second the opposite way. This is because the marked cells could differ depending on the inclination of the segment, as shown in Fig. 6. Note that, the probability decrement value,  $\tau_d$ , is lower than the probability increment value,  $\tau_i$ . This is due to two main reasons. First, if the cell is not occupied but its probability has been increased due to noisy measurements, this probability will be low because noisy measurements are usually formed by scattered points. And second, as the represented data depends on the height interval, it is possible that nearby short objects do not block the line of vision to high objects behind them. The only problem of this method is that dynamic objects located in cells with high probability will need several iterations of the mapping algorithm to be completely updated.

### 3.4. Creation of the visualization of the map

After updating the occupancy probability of all the cells, the visual representation of the map is created using a gray-scale image where each pixel corresponds to a cell. Therefore the size of the image depends on the number of grids created to represent the environment. The value of each pixel is directly calculated from the occupancy probability of each pixel. The darker cells possess the

higher probabilities. Obviously, a white pixel will represent a non-occupied cell.

Once the image is created, an opening morphology operation is applied to achieve a more structured-looking and visually friendly map as shown in Fig. 7. The opening of an image  $A$  by the structuring element  $B$  is obtained as follows:

$$A \circ B = \bigcup_{B_x \subseteq A} B_x \quad (7)$$

where  $B_x$  is the structuring element  $B$  translated to localization within  $A$ .

It is worth explaining how different objects are represented in the map. In Fig. 7 the walls of the room are represented in two ways, namely, as a single thin line (the horizontal wall) and as a cloud of points (the vertical walls) due to the observed objects. In the horizontal wall, only the wall and a door are observed, resulting on a clear representation in the map. But in the two vertical walls, several objects are registered at different distances. Higher objects (within the height interval) behind shorter objects are gotten and, therefore, the map shows obstacles in different points resulting in a non-uniform representation.

## 4. Data and results

In this section, the results and configuration of the experiments are presented. For our experiments, two different indoor environments are considered. The first test environment consists of a hall of approximately  $6 \times 8$  m, located between a wide corridor and downward stairs, as shown in Fig. 8. In the second experiment, the algorithm is tested in a laboratory with several obstacles (furniture, boxes, etc.) as shown in Fig. 12c. The ToF camera used in the research and experiments is a Mesa Imaging camera model SR4000.

### 4.1. Results for the hall scenario

In a first experiment the mapping algorithm has been tested in an environment where the ToF was believed to have some problems. The main difficulties for the camera at this experiment are: (a) objects at distances above the camera's measurement range, (b) light reflections due to metallic and glass surfaces, and, (c) sources of intense light present in the scene. These three problems influence the performance of the ToF camera, but the results show that our algorithm is robust against them. In the test environment, several doors and a few objects can be found, but it is mostly a simple environment which should result in a clear map representation.

The generated map is presented in Fig. 9. As you may note the map is composed of three grids, dynamically created when new data is collected beyond the already existing grids.

Fig. 10 shows a correspondence of the real map and the created one. Note that it mainly corresponds to the initial grid. Both are represented in the same scale, measured in meters, where point (0,0) is the initial position of the robot.

Some details in Fig. 10a have to be explained. The map shows pixels with different gray levels. Therefore the occupancy probability of the cells varies from one to another pixel. A lower gray level generally means that less data has been acquired by the camera within that cell. This is caused by the performance of the ToF camera when recording light sources or reflective objects (such as glass or metallic surfaces). In these cases, the intensity of the signal folded back to the ToF camera is too high and saturates it, making the camera discard that information. Obviously, the dismissed information is not added to the map, and the occupancy probability of the corresponding cells are not increased. This error can be



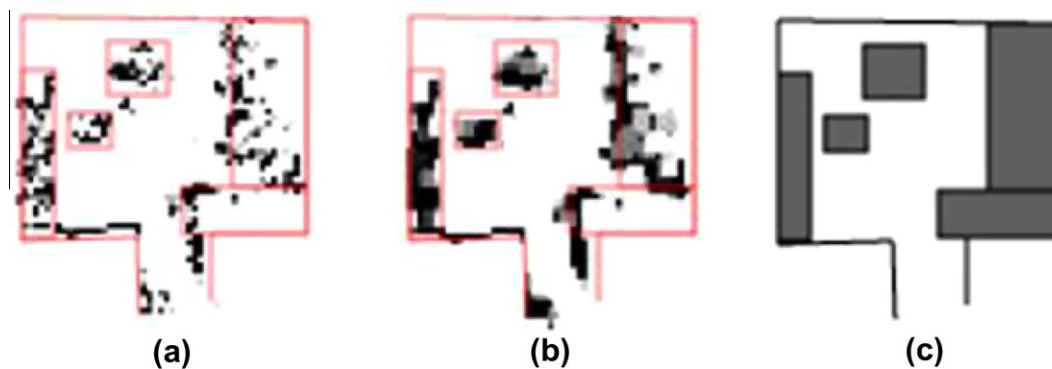


Fig. 7. Opening morphology operation on the resulting map. (a) Original resulting map. (b) Resulting map after opening operation. (c) Mapped room.

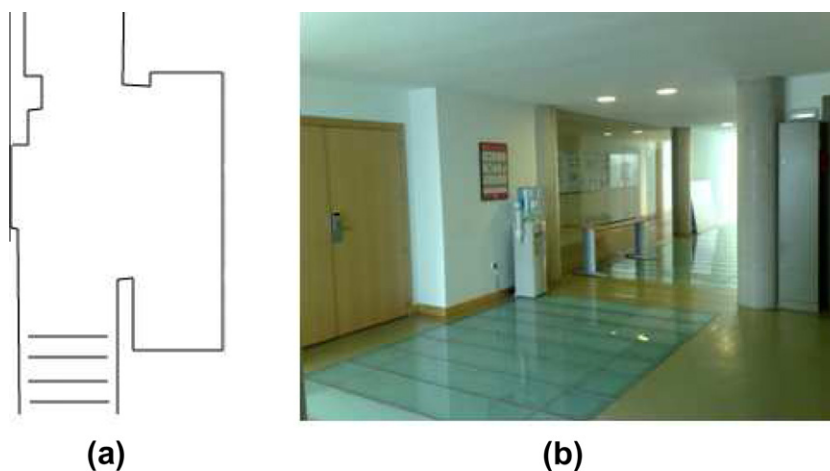


Fig. 8. Hall test scenario. (a) Map of the environment. (b) A partial view of the environment.

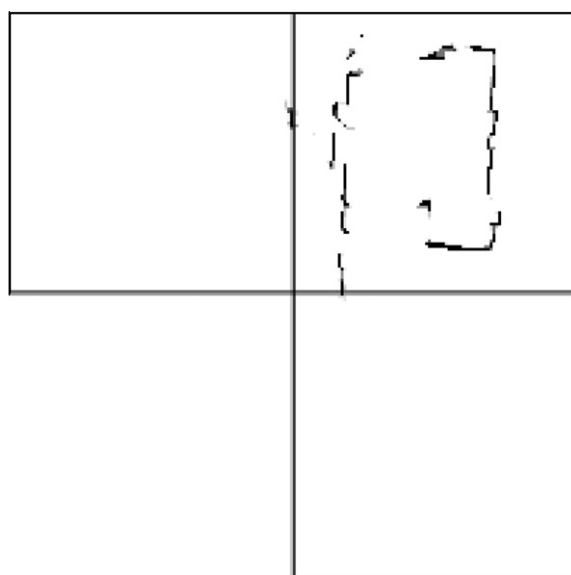


Fig. 9. Generated map divided into three grids. The top right grid is the starting grid.

solved by changing the angle of observation between the robot and the reflective object. This problem is depicted in Fig. 10aa, where a metallic door of an elevator at coordinates (1,4) results in a discontinuity in the wall of the generated map. Another example is

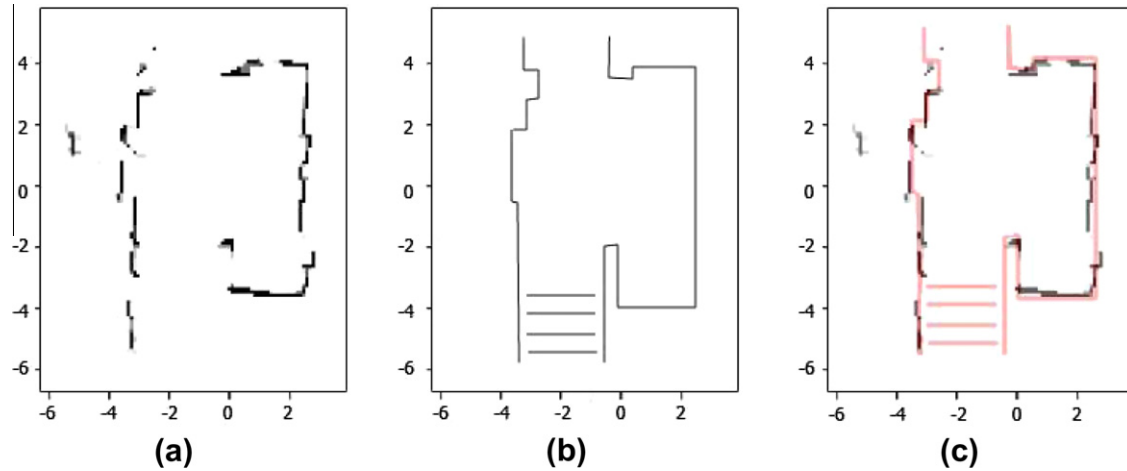
detected at  $(-3, -3)$  due to the capture of the glass handrail of the stairs. Moreover, two differences between the generated map and the real map may be observed at  $(2, -2)$  and  $(-4, 2)$  coordinates. The first one corresponds to a partially open door, and the second one is an open door with an object observed inside the adjacent room placed in coordinates  $(-5, 2)$ . Finally, the right wall of the stairs is not included in the map because it was never in the field of vision of the robot.

The map shown is obtained after 23 iterations of the mapping algorithm. Fig. 11 presents an example of the map's evolution between the third and fourth of the intermediate iterations. Fig. 11a shows the map after the third iteration and Fig. 11b after the fourth iteration. A zoom of the upper right part of the map is shown. The exact correspondence with the final generated map in Fig. 10 is given by the coordinates expressed in meters – as mentioned above, the initial position of the robot corresponds to coordinates  $(0, 0)$ . Comparing the segment between  $(2.25, 3.5)$  and  $(2.25, 2)$  in Fig. 11a and b, it is visible that data collected in both iterations overlap. As shown, the gray level of some pixels is higher after the second iteration, which means that the occupancy probability of previously filled cells increases when new data is detected.

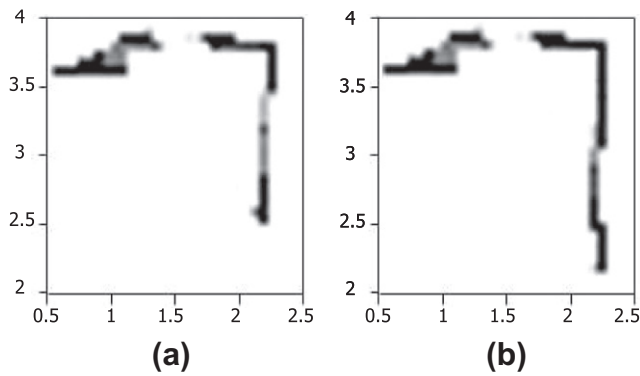
The environment used in this experiment is quite stable as there are not many objects to represent. As a result, the generated map contains clear contours.

#### 4.2. Results for the laboratory scenario

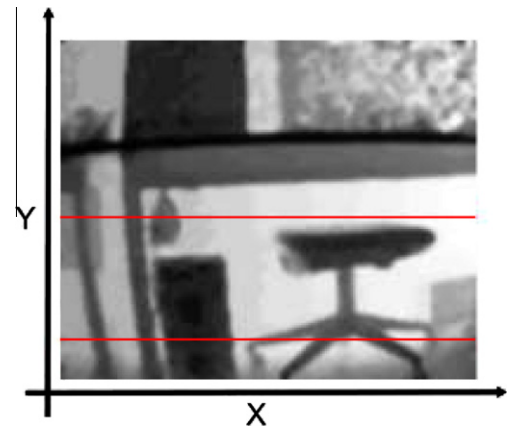
In the second experiment, the SLAM algorithm is tested in a laboratory where several obstacles are present. Fig. 12a shows



**Fig. 10.** Comparison of the generated map and the environment map in a coordinate system expressed in meters. (a) Generated map. (b) Real map. (c) Generated and real maps overlapped.



**Fig. 11.** Updating of the occupancy probability in successive iterations. (a) Iteration number 3. (b) Iteration number 4.

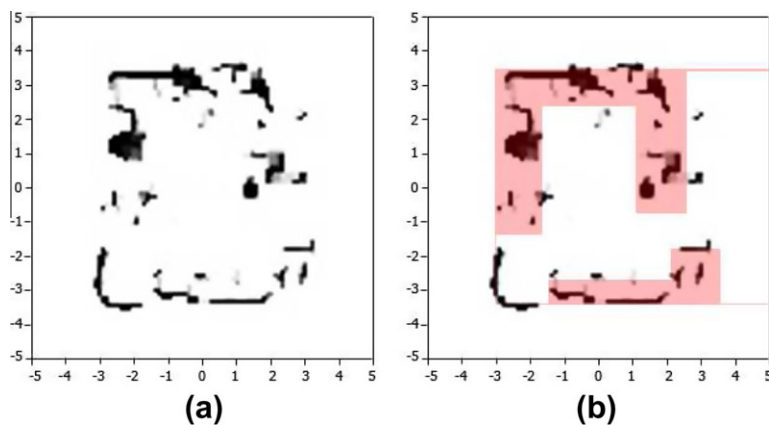


**Fig. 13.** Different obstacles detected with the selected height interval.

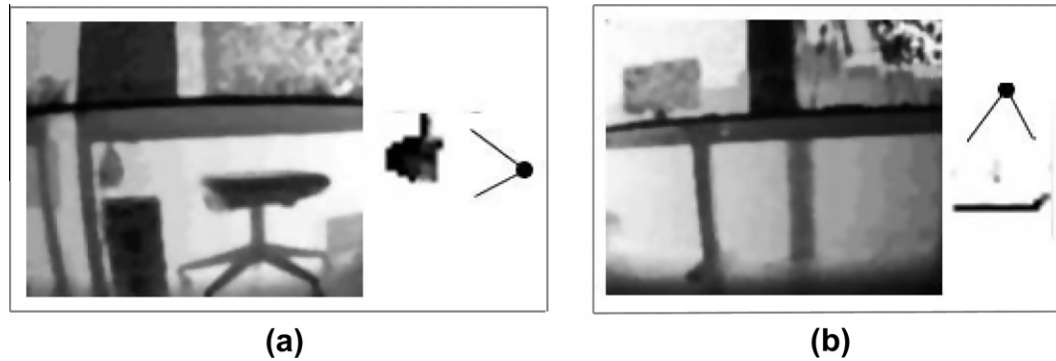
the generated map, and the grid's coordinate system expressing distances in meters, being (0,0) the initial robot position. In Fig. 12b, zones highlighted with red color<sup>1</sup> represent areas where obstacles are placed that, as Fig. 12c shows, are mainly composed of tables and several objects under them. Also note the presence of chairs in the scenario which is also reflected in the results.

There are two factors which have complicated the representation of the generated map. First, the presence of the aforementioned obstacles in the scenario, and second, the height interval

that determines which objects will be included in the map's representation. Fig. 13 presents the ToF observation made at a random iteration. It is presented with the approximate height interval used in the experiment. In the figure, it is shown that several objects (including the wall) are found within the same height interval and, therefore, included in the map. If just one obstacle is detected for a given  $x$  position, this is represented as a single darker pixel (as it happened in the first test where no objects were placed in the



**Fig. 12.** Laboratory test scenario. (a) Generated map. (b) Generated map where the areas of the obstacles are highlighted. (c) Partial view of the environment.



**Fig. 14.** Comparison of the representation of different kinds of obstacles. (a) Left: ToF image of several obstacles at different heights. Right: representation in the map (coordinates  $(-2, 1)$ ) and position of the robot. (b) Left: ToF image of a single obstacle (a wall). Right: representation in the map (coordinates  $(1, -3)$ ) and position of the robot.

scene). But when different obstacles are in the same  $x$  position (for example, a taller object behind a shorter one), the representation of the map seems more abstract because all of them are included with different occupancy probabilities.

To understand the difference between the representation of a group of obstacles in the same direction and the representation of only one obstacle (the most intuitive case would be a wall), Fig. 14 shows a comparison of the representation of multiple objects (see Fig. 14a), and the representation of a wall (as shown in Fig. 14b). The figure also presents the ToF images and the relative position of the robot.

## 5. Conclusions

In this paper, a map building algorithm for mobile robots has been introduced. The perceived environment is represented in a map containing in each cell a probability of presence of an object or part of an object. As the size of the environment is unknown a priori, it is not possible to create a fixed-size occupancy grid. The environment is represented as a collection of modular occupancy grids which are added to the map as far as the mobile robot finds objects outside the existing grids.

In our approach a ToF camera is exploited as a range sensor for the mapping purpose. Indeed, a ToF camera used as a range sensor provides a powerful tool for detecting objects in front of the robot by measuring the distance towards them. ToF cameras provide depth information per pixel which make them ideal for background foreground segmentation, as in general the depth defines the subject from background in a much more basic way than the light intensity does. Thus, one of the areas where ToF sensors are adequate is in obstacle avoidance, because the detection region is not only horizontal but also vertical, allowing to detect obstacles with complex shapes. Also, thanks to the larger vertical field of view of ToF cameras, difficult obstacles are better detected by a ToF camera than by a 2D laser scanner.

The main steps of the map building algorithm are extensively described in the paper. First, the pixels' distance information is selected, where the height interval is chosen according to the purpose of the map. After selecting the pixels' height range to create the map, their positions in the map are calculated. Once the coordinates of every point have been calculated, the new observations must be represented in the grid, updating the occupancy probability of the affected cells. As each cell covers several squared centimeters of the map, it is usual that more than one detected point is located into the same cell for the same observation. Consequently, the occupancy probability of each cell is increased as many times as points are detected to belong to it. After updating the occupancy probability of all the cells, the visual representation

of the map is created using a gray-scale image where each pixel corresponds to a cell. Therefore the size of the image depends on the number of grids created to represent the environment. The value of each pixel is directly calculated from the occupancy probability of each pixel.

The map building algorithm has been successfully tested in two different and complex indoor environments. The next step in our research is to use the generated maps for autonomous navigation.

## Acknowledgements

This work was partially supported by the Spanish Ministerio de Ciencia e Innovación under Project TIN2010-20845-C03-01, and by the Spanish Junta de Comunidades de Castilla-La Mancha under Projects PII2I09-0069-0994 and PEII09-0054-9581.

## References

- Begum, M., Mann, G. K. I., & Gosine, R. G. (2008). Integrated fuzzy logic and genetic algorithmic approach for simultaneous localization and mapping of mobile robots. *Applied Soft Computing*, 8, 150–165.
- Benet, G., Martínez, M., Blanes, F., Pérez, P., & Simó, J. E. (2005). Differentiating walls from corners using the amplitude of ultrasonic echoes. *Robotics and Autonomous Systems*, 50, 13–25.
- Chatterjee, A., & Matsuno, F. (2010). A Geese PSO tuned fuzzy supervisor for EKF based solutions of simultaneous localization and mapping (SLAM) problems in mobile robots. *Expert Systems with Applications*, 37(8), 5542–5548.
- Chatterjee, A., Ray, O., Chatterjee, A., & Rakshit, A. (2011). Development of a real-life EKF based SLAM system for mobile robots employing vision sensing. *Expert Systems with Applications*, 38(7), 8266–8274.
- Davison, A. J., Reid, I., Molton, N., & Stasse, O. (2007). MonoSLAM: real-time single camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6), 1052–1067.
- Eade, E., & Drummond, T. (2009). Edge landmarks in monocular SLAM. *Image and Vision Computing*, 27, 588–596.
- Feng, H. M., Chen, C. Y., & Horng, J. H. (2010). Intelligent omni-directional vision-based mobile robot fuzzy systems design and implementation. *Expert Systems with Applications*, 37(5), 4009–4019.
- Fernández-Caballero, A., Castillo, J. C., Serrano-Cuerda, J., & Maldonado-Bascón, S. (2011). Real-time human segmentation in infrared videos. *Expert Systems with Applications*, 38(3), 2577–2584.
- Fernández-Caballero, A., López, M. T., & Saiz-Valverde, S. (2008). Dynamic stereoscopic selective visual attention (DSSVA): Integrating motion and shape with depth in video segmentation. *Expert Systems with Applications*, 34(2), 1394–1402.
- Foix, S., Alenyá, G., & Torras, C. (2010). *Exploitation of time-of-flight (ToF) cameras*. IRI Technical report IRI-TR-10-07. Institut de Robòtica i Informàtica Industrial.
- Gascuña, J. M., & Fernández-Caballero, A. (2011). Agent-oriented modeling and development of a person-following mobile robot. *Expert Systems with Applications*, 38(4), 4280–4290.
- Gemeiner, P., Jovic, P., & Vincze, M. (2009). Selecting good corners for structure and motion recovery using a time-of-flight camera. *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems*, 5711–5716.
- Goedemé, T., Nuttin, M., Tuytelaars, T., & Van Gool, L. (2007). Omnidirectional vision based topological navigation. *International Journal of Computer Vision*, 74, 219–236.



- Guethmundsson, S. A., Pardas, M., Casas, J. R., Sveinsson, J. R., Aanaes, H., & Larsen, Rasmus (2010). Improved 3D reconstruction in smart-room environments using ToF imaging. *Computer Vision and Image Understanding*, 114, 1376–1384.
- Hedge, G., & Ye, C. (2009). Extraction of planar features from Swissranger SR-3000 range images by a clustering method using normalized cuts. *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems*, 4034–4039.
- Kaess, M., & Dellaert, F. (2010). Probabilistic structure matching for visual SLAM with a multi-camera rig. *Computer Vision and Image Understanding*, 114, 286–296.
- Lange, R. (2000). *3D Time-of-flight distance measurement with custom solid-state image sensors in CMOS/CCD-technology*. Ph.D. Dissertation, University of Siegen, Germany.
- López, M. T., Fernández-Caballero, A., Fernández, M. A., Mira, J., & Delgado, A. E. (2007). Dynamic visual attention model in image sequences. *Image and Vision Computing* (2007), 597–613.
- López-Valles, J. M., Fernández, M. A., & Fernández-Caballero, A. (2007). Stereovision depth analysis by two-dimensional motion charge memories. *Pattern Recognition Letters*, 28(1), 20–30.
- Marks, T., Howard, A., Bajracharya, M., Cottrell, G., Matthies, L. (2007). Gamma-SLAM: stereo visual SLAM in unstructured environments using variance grid maps. In: *IROS visual SLAM workshop*.
- May, S., Droschel, D., Holz, D., Wiesen, C., Fuchs, S. (2008). 3D pose estimation and mapping with time-of-flight cameras. In *Proceedings of the IEEE/RSJ IROS workshop on 3D-mapping*.
- May, S., Droschel, D., Holz, D., Fuchs, S., Malis, E., Nüchter, A., et al. (2009). Three-dimensional mapping with time-of-flight cameras. *Journal of Field Robotics*, 26(11-12), 934–965.
- Moreno, F. A., Blanco, J. L., & Gonzalez, J. (2009). Stereo vision specific models for particle filter-based SLAM. *Robotics and Autonomous Systems*, 57, 955–970.
- Paz, L. M., Piniés, P., Tardós, J. D., & Neira, J. (2008). Large scale 6DOF SLAM with stereo-in-hand. *IEEE Transactions on Robotics*, 24(5), 946–957.
- Schleicher, D., Bergasa, L. M., Ocaña, M., Barea, R., & López, E. (2010). Real-time hierarchical stereo visual SLAM in large-scale environments. *Robotics and Autonomous Systems*, 58, 991–1002.
- Sim, R., Elinas, P., & Little, J. J. (2007). A study of the Rao–Blackwellised particle filter for efficient and accurate vision-based SLAM. *International Journal of Computer Vision*, 74(3), 303–318.
- Sim, R., & Little, J. J. (2009). Autonomous vision-based robotic exploration and mapping using hybrid maps and particle filters. *Image and Vision Computing*, 27, 167–177.
- Weingarten, J. W., Gruener, G., & Siegwart, R. (2004). A state-of-the-art 3D sensor for robot navigation. *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems*, 3, 2155–2160.
- Williams, B., Cummins, M., Neira, J., Newman, P., Reid, I., & Tardós, J. (2009). A comparison of loop closing techniques in monocular SLAM. *Robotics and Autonomous Systems*, 57, 1188–1197.
- Wurm, K. M., Stachniss, C., & Grisetti, G. (2010). Bridging the gap between feature- and grid-based SLAM. *Robotics and Autonomous Systems*, 58, 140–148.