Hindawi
Mathematical Problems in Engineering
Volume 2019, Article ID 2605837, 13 pages
https://doi.org/10.1155/2019/2605837

Research Article

An OGM-Based Visual Navigation System with Reduced Homographies and Accurate Computations for Mobile Robots

Erick I. Guerra-Hernandez,[1] **Mario A. Ibarra-Manzano**,[1] **Patricia Batres-Mendoza**,[1] **Dora L. Almanza-Ojeda**,[1] **Jorge A. Soria-Alcaraz**,[2] **Andres Espinal**,[2] **and Horacio Rostro-Gonzalez**[1]

[1]*Department of Electronics Engineering, University of Guanajuato, 36885 Salamanca, Gto., Mexico*
[2]*Department of Organizational Studies, University of Guanajuato, 36250 Guanajuato, Gto., Mexico*

Correspondence should be addressed to Horacio Rostro-Gonzalez; hrostrog@ugto.mx

Received 22 February 2019; Accepted 3 June 2019; Published 24 June 2019

Academic Editor: Ning Sun

This article presents a novel approach to calculate and accelerate the Occupancy Grid Mapping (OGM) for real-time mobile robot navigation. Here, we have improved the main process involved in the construction of OGMs in two ways. On one hand, we propose a new method to perform the camera calibration (CC) process, in which we reduce the number of intermediate steps (homographies) when the reference system is transformed from the real world into a navigation map, reducing consequently the computational costs. On the other hand, the OGM is constructed by an online probabilistic method, which simultaneously performs both the camera calibration and maps construction, thus reducing cumulative errors when these two processes are separately treated. Furthermore, the proposed system can be easily parallelized and mapped onto a digital system (e.g., Field Programmable Gate Arrays, FPGAs) for real-time robot navigation.

1. Introduction

Robot navigation is an important task that allows robots to perform activities such as localization, path planning, collision avoidance [1], and people finding [2]. Here, environment modeling plays a key role and several methods have been proposed, e.g., Occupancy Grid Mapping (OGM) [3], Geometric Mapping (GM) [4], and Topological Mapping (TM) [5], with the former being the predominant one. The OGM method creates maps from environments which have been sensed by either a sonar [3, 6] or a different sensor such as laser range finder (e.g., Laser Imaging Detection and Ranging (LIDAR)) [7–11], ultrasonic [12], radar [13], and vision [14–18], as well as using a combination of them [19–24]. Thereby, any environment can be mapped using OGM-based methods, albeit estimating the localization of a robot based on its position and orientation in a mapped environment with a high-resolution may require high computational costs [25].

Traditional OGM-based methods have some issues, such as a wrong assignation of occupancy probabilities in maps during the sensor readings, specially when its beam is reflected over several surfaces and the beam may return or not. Another problem occurs during the mapping when grid-elements' probabilities reach a probability of 1 and further sensor's readings are not capable of changing them [26]. In this work, two major contributions on the main processes associated with OGM-based visual navigation systems; that is, Camera Calibration (CC) and Occupancy Map construction are presented. In the former, we extract accurate metric information [27, 28] from the environment, which serves as input to generate the OGM and navigation system. On the occupancy map construction, spatial representations of the robot environment [2] are created by covering the plane with repetitions of a particular regular polygon (tessellation of the plane) [29] that stores a probabilistic estimation of its state (empty or occupied). To the best of our knowledge, these two processes are usually treated separately; thus, one main contribution of this work is that such processes are simultaneously implemented and performed to build up the proposed OGM-based visual navigation system. The other main contribution of this work is regarding the CC method; there are several CC methods and different classifications (see [30, 31] for comprehensive reviews), where its fundamental operation is the homography, usually represented as a 3 × 3 matrix and denoted by H [32]. This is an invertible mapping from a projective plane to itself such that three points lie on the same line if and only if their mapped points are also collinear [33, 34]. Herein, the proposed CC method uses only two homographies, fewer than a traditional CC method (four in most of the cases), thus reducing the high computational costs and speeding up the process when the OGM is calculated.

In Figure 1, we graphically highlight the improvements of our approach to perform the CC process compared with a traditional one. In such figure, red and green arrows correspond to homography operations required by our approach and those in traditional ones, respectively. Here, the CC method allows associating points from the real-world's $\{W\}$ to grid-elements of the OGM $\{G\}$. The traditional CC method, green arrows labeled as TM (Traditional Method) from 1 to 4, calculates the homography H^{wr} transforming the real-world's reference system $\{W\}$ into the robot's reference system $\{R\}$ in the step TM1; next the step TM2 transforms from robot's reference system $\{R\}$ into camera's reference system $\{C\}$ by using the homography H^{rc}; afterwards the step TM3 transforms camera's reference system $\{C\}$ into the camera's plane of the image $\{I\}$ (converting meters to pixels) by means of homography H^{cf}; finally, homography, H^{fm}, transforms the camera's plane of the image $\{I\}$ into the OGM $\{G\}$ (see [30] for implementation details). The proposed CC method, red arrows labeled as PM (Proposed Method) from 1 to 2, is as follows: step PM1 transforms from the real-world's reference system $\{W\}$ into the robot's reference system $\{R\}$ by means of an homography H^{wr} and finally PM2 through a second homography H^{rm} directly transforms from the robot's reference system $\{R\}$ into the OGM $\{G\}$; this is achieved by means of a particular experimental configuration of the calibration pattern (chessboard-like proposed in [27]) and the camera sensor, later explained.

The proposed system aims to tackle some of the main problems associated to OGM-based visual navigation systems, such as CC methods being computational intensive and there is a trade-off between the size and the resolution of the grid, so that to fulfill the real time requirements it is common practice to use a simplified inverse sensor model that maps the measurements to the grid [24]; this result affects the precision of the representation [35]. In the proposal, the CC method and OGM construction are both together treated and performed, contrary to most works that commonly process them separately. Besides, the grid-elements of OGM are generated from the calibration pattern which favors to dispense sensor model generation and allows to reduce cumulative errors derived from rounded values of homographies. Besides, the system was designed to be implemented on digital systems such as Field Programmable Gate Arrays (FPGAs) to take advantage of their capabilities to parallelize algorithms.

The rest of the paper is organized as follows: Section 2 presents some related works. Section 3 introduces the proposed OGM-based visual navigation system including its components and methods. Numerical experiments, configuration, and results are shown in Section 4. Finally, in Section 6, conclusions of this work and future work are given.

2. Related Works

Maps are an interface for the visualization of environmental data that allows the access to information and exploratory activities [36]. In this regard, the OGM is a method for modeling both indoor and outdoor environments through a 2D or 3D grid, in which each grid-element stores quantitative information about the occupancy state (empty or occupied) of the environment's region it covers. The creation of an OGM is not a trivial task, and it requires translation of the data coming from sensors by means of sensory models to make estimations about the status of the environment being modeled [26]. The OGM method was introduced in [3] and it consists of radar sensors and implements two intermediate models to map the probabilities of empty and occupied regions sensed from the environment; these models are unified according to an updating rule where the greater probability value from the same spatial grid-element between two models is chosen to define the occupancy state (unknown, empty, or occupied) of the corresponding grid-element in the unified or global map. Later, in [37] a modification to update the global map was proposed, which is based on Bayes Theorem. However, these two updating rules have numerical issues when probabilities tend to extreme values of 0 (empty) or 1 (occupied) [38]; this issue was overcame by Thrun's proposal, where the occupancy state of each grid-element is calculated as the logarithmic value of probability of being occupied divided by the probability of being empty [39].

Since their creation the OGM methods have been the dominant paradigm to model environments; nevertheless, it is still a highly active research field. Several proposals to generate OGMs using different kind of sensors have been published, although recently most of researches have been using visual sensors such as CCD and RGB-D cameras. Next reviewed works are described.

The LIDAR sensors have been used in several works for capturing environment information and modeling through OGM. In [7], information is processed by assigning different probabilistic functions to each occupied grid-element. A Simultaneous Localization and Mapping (SLAM) task is carried out in [8], and this is made through a binary Bayesian filter as updating function, besides mutual information, to predict the uncertainty degree of each grid-element. In [9], a SLAM task is performed at hardware level by using genetic algorithms and updating just the grid-elements along the exploration area of the sensor. With the purpose of reducing hardware requirements, in [10] there are several sensors integrated that uniquely use integer arithmetic in order to represent the probability of grid-elements. A novel sensor model to map and detect moving objects is proposed in [11]. The LIDAR and radar sensors are used besides a Graphic

Processing Unit (GPU) to generate maps and an approach based on histograms is introduced for detection of highways limits in [24].

Biological vision systems are a powerful tool to explore and gather information from the environment. These systems have inspired the development of vision cameras such as CCD and RGB-D, which have been widely used for different applications, e.g., obstacle detection and region segmentation [40]. In [20], there are generated 3D probabilistic OGM for the autonomous navigation of aerial vehicles. A system for automatic driving of cars in highways is proposed in [14]; the system processes 2D and 3D images and uses disparity maps to estimate the allowed highways sections to drive. Problems with moving obstacles are studied in [16, 23]; these works used particle-based OGM to detect the objects; each article has associated a position and velocity. In [17, 18], moving objects are detected through OGMs generated by means of stereo vision and a sigmoid-function-based updating rule. An automatic exploration method is proposed in [21], and this is based on the entropy change in order to avoid the future uncertainty in the map. Finally, in [19] a SLAM task was carried out by means of information extracted from structural data from architectonic blueprints with the purpose of seek and rescue applications.

Most of aforementioned works implement Thrun's updating rule due its versatility to deal with a diversity of sensors. Particularly, OGM-based visual navigation systems provide a large quantity of information over the environment when they perform a certain task, and this is why we decided to design a specific updating rule that deals with such kind of systems. Also, regarding the difference of traditional OGM generation methods such as those proposed in [7–11, 14, 16–19, 21, 23, 24] our proposal is capable of detecting small obstacles. Albeit our paper presents advantages over traditional OGM-based methods, its main drawback is the long time it takes to process such amount of data.

3. OGM-Based Visual Navigation System

Our proposal is schematized in Figure 2. This is mainly build up by three functional blocks, i.e., acquisition and image enhancement (green block), camera calibration (yellow block), and OGM construction (blue block).

3.1. Acquisition and Image Enhancement (AIE). This block provides the input images to CC and OGM construction blocks of the system. This block is defined by two modules: the Camera Module that captures images from robot's environment and the Gamma Correction Module which enhances the image's contrast to be processed or handled by further blocks. Figure 3(a) shows a sample of an original image captured by the camera module and Figure 3(b) shows the same image after enhancement with gamma correction module. Next, each module is described.

3.1.1. Camera Module. In this work, we considered a highly portable EVB1005 CCD camera from Opal Kelly, Figure 4(a), with the following technical specifications: an Aptina MT9P031 5-Mpixel CMOS image sensor, 14 fps at full resolution (2592 x 1944), 96 MHz pixel clock, 12-bit ADC resolution. Also, this camera can be easily connected to a ZEM4310 from Opal Kelly, which is based on the Altera Cyclone IV FPGA (Figure 4(b)) for deployment of real-time embedded robotic systems and high level IO communication.

3.1.2. Gamma Correction Module. In our approach, images were captured in a noncontrolled environment by a camera with a low aperture's shutter; for this reason they present illumination variations. Thus, to enhance the colorimetric information of such images a gamma correction factor is applied. The factor is given by the following equation:

$$g(u) = u^{\gamma} \tag{1}$$

where $u \in [0,1]$ denotes the image pixel intensity and we define the correction factor γ in 0.65 (see [41] for more details).

3.2. Camera Calibration (CC). The CC block carries out the namesake main process of OGM-based visual navigation systems. Here, camera calibration is performed to extract metric information [27, 28] and finally construct a grid required to build the OGM for mobile robot navigation. Figure 5 schematizes the chessboard-like pattern resulting from CC process. The CC block assigns an index to label each pixel of an image according to the grid-element it belongs to and counts how many pixels are in each grid-element. For the herein proposed CC method, the camera sensor and the calibration pattern (a black and white chessboard-like pattern [27] with squares of size 10 × 10 cm) are prefixed; the camera is set over the robot, while the calibration pattern is set on the floor and in front of the camera; this configuration allows the camera's intrinsic parameters to be considered as constants. Thus only H^{wr} and H^{rm} homography operations are required to directly transform from real-world's reference system $\{W\}$ into the OGM $\{G\}$ instead of using several intermediates homographies as a traditional CC method generally requires (see Figure 1).

Despite the fact that the CC block has the highest computational cost in the whole proposed system, the calculations and data acquirement are performed once. Besides, the proposal does not require to know the region that each pixel occupies per geometric projection, but per each grid-elements and the number of pixels and their spatial position that it contains are required. The grid-elements closer to the camera have a higher definition compared with those faraway.

The CC process is performed offline by the following modules: Harris Corner Detector, Definition of Grid-Element's Components, and Grid-Element Pixel Counting.

3.2.1. Harris Corner Detector Module. It implements the Harris algorithm [42] to detect corners in the captured and enhanced images by the AIE module. This allows, either, to add missed corners or to remove those wrongly detected. The Harris' algorithm returns the set of coordinates $S = \{(x_1, y_1), \ldots, (x_N, y_N)\}$ where the N corners were detected.

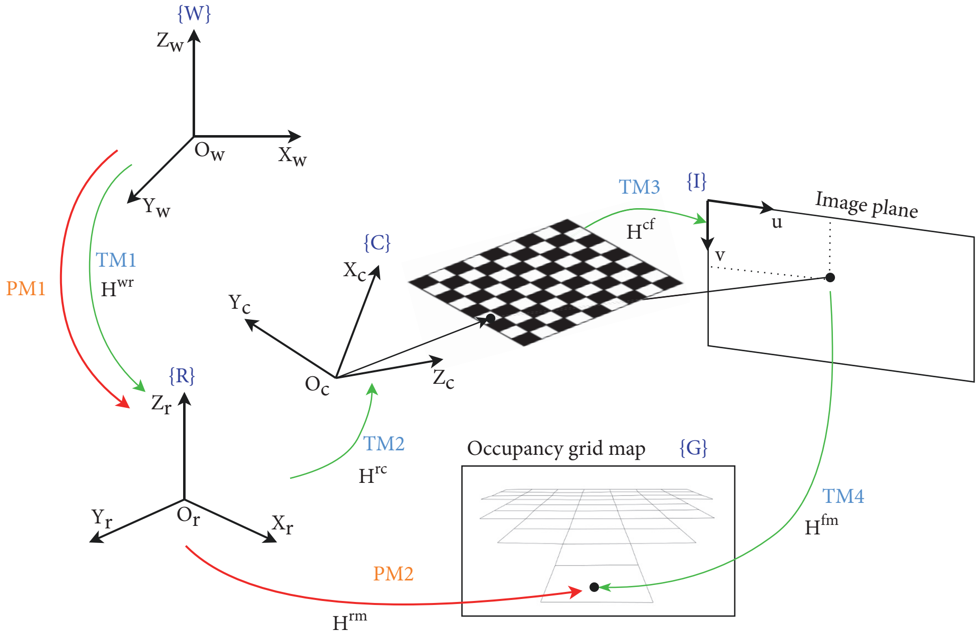


Figure 1: A comparative diagram showing the required homography operations in the traditional (green arrows) and proposed (red arrows) CC methods (based on [30]).

Table 1: Rules for grid update of the global map.

Global map	Local map	Global map
Previous probability	Current probability	Current probability
$O_{GM}^{ge}(k-1)$	$O_{LM}^{ge}(k+1)$	$O_{GM}^{ge}(k)$
free	free	$O_{LM}^{ge}(k)$
free	unknown	$O_{GM}^{ge}(k-1)$
free	occupied	$P_R * O_{LM}^{ge}(k) + Ph * O_{GM}^{ge}(k-1)$
occupied	free	$P_R * O_{LM}^{ge}(k) + Ph * O_{GM}^{ge}(k-1)$
occupied	unknown	$O_{GM}^{ge}(k-1)$
occupied	occupied	$O_{LM}^{ge}(k)$
unknown	free	$O_{LM}^{ge}(k)$
unknown	unknown	$O_{GM}^{ge}(k-1)$
unknown	occupied	$O_{LM}^{ge}(k)$

In Figure 6, the detected corners in the calibration pattern are shown.

3.2.2. Definition of Grid-Element's Components Module. In this module, images are segmented in order to define the grid-elements of the chessboard-like pattern. *For this purpose, it is necessary to group the grid corners of the calibration image in rows and columns as shown in Figure 7, where the corners are represented by yellow squares and organized in horizontal lines (in blue) and vertical lines (in red).* For each horizontal line, its first element would be the most-left corner, whereas the upper one in each vertical imaginary line would be considered the first one. In the same manner, grid-elements can be defined by grouping rows and columns taking the first element similarly as what has been made for imaginary lines. This module is performed through the next four steps.

(1) *Horizontal Corners Grouping.* The objective is to identify which corners of S are part of the same horizontal line. For this, the Euclidean distances between the first corner of a horizontal line and the subsequent corners are calculated, considering only the components in the y-axis; that is, the pairs of corners whose euclidean distance is less than a heuristic value k (see (2)):

$$d\left(\left(0, y_i\right), \left(0, y_j\right)\right) < k \qquad i = 1, \ldots, N-1 \text{ and } j = i+1, \ldots, N \tag{2}$$

(2) *Vertex's Grid-Element Estimation.* To estimate the vertices that conforms a grid-element, the first corner from two contiguous horizontal lines must be in the

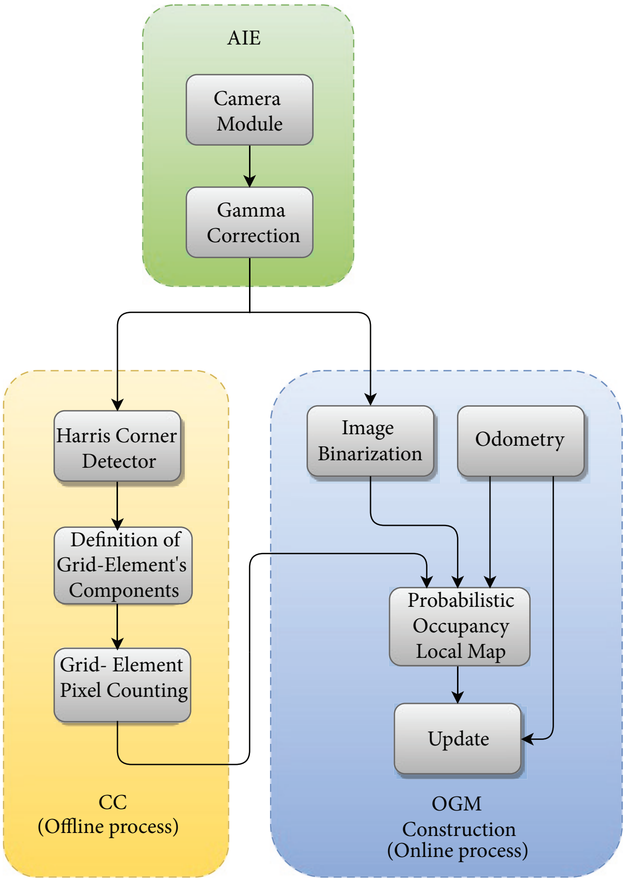


Figure 2: Block diagram of the proposed OGM-based Visual Navigation System.

same vertical line. This is verified by calculating and comparing the Euclidean distances d_1 and d_2 (see Figure 8); the former distance is calculated by using the first corner $S(1, r)$ on a horizontal line r and the first corner $S(1, r+1)$ on the next horizontal line, and then d_2 is calculated by using the second corner $S(2, r)$ and the same $S(1, r+1)$. Besides, (3) is used to figure out which detected corner is the left side of a grid-element; that is, if the left side of equation is greater than the right side then the left side of a grid-element is in the same row r rather than the corner $S(1, r)$ (see Figure 8(a)). Otherwise, the left side of a grid-element is in the same column rather than the corner $S(2, r)$ (see Figure 8(b)). This is used to identify grid-elements through its conforming vertexes:

$$d(S(1,r), S(1,r+1)) < d(S(2,r), S(1,r+1)) \tag{3}$$

(3) *Line Segments Definition*. By using the two-point form of the equation of a line, we define the segments that form each grid-element; that is, the slope and y-intercept point are calculated.

(4) *Grid-Elements Labeling*. Finally, each pixel is assigned a label with the number of the grid that contains it. Labeling is performed by evaluating the y-intercept coordinates of the lines with the value in the x-component position of the pixel. Then, the y-intercept coordinate with the highest value is used to determine the pixel's position with respect to the four sides of each calculated grid allowing finding out if the pixel is contained in any grid; if so a label according to the number of grid-element is assigned to the pixel; otherwise a zero value is assigned.

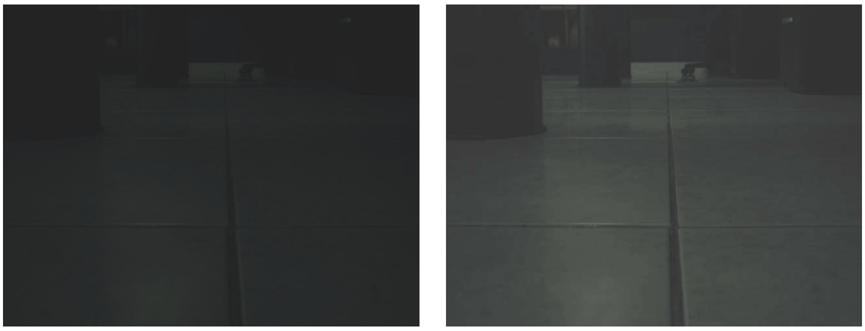
The result of these four steps is shown in Figure 9; here we can observe the chessboard-like pattern with the segmented and labeled grid-elements marked with yellow color.

3.2.3. Grid-Element Pixel Counting Module. This module has been designed to count the pixels of each grid-element over the whole grid according to the last step in the Definition of Grid-Element's Components. *The resulting information is stored in a hash table and it is later used besides Imagen Binarization module's output in order that the POLM module can estimate the occupancy probability of grid-elements. A detailed description is presented in the next section.*

3.3. Occupancy Grid Maps Construction. The global mapping corresponds to the exploration of the whole environment and in turn is built up by several Probabilistic Occupancy Local Maps (POLMs). In the beginning, an unknown environment has a OGM with a general occupancy probability that is equal to 0.5, which corresponds to an unknown occupancy state for all POLMs and its grid-elements, according to (4). To sense an environment through a POGM that requires generating several POLMs, the whole process is as follows (see Figure 10). The first step is performed by the Image Binarization Module which segments images captured in AIE block to distinguish between floor and obstacles regions. Then, due to the lack of knowledge about spatial and orientation information of the current POLM and the general OGM, an odometry process is performed as a second step. The Probabilistic Occupancy Local Map module, in the third step, generates a partial map according to the orientation of the camera, which will be used to define and update the OGM. The fourth and last step is the Update Module, which updates the OGM according to its previous occupancy states and the current occupancy states of its POLMs. These modules serves to either create a OGM of an unknown environment or update one previously mapped:

$$OS_{GM}^{ge}(K) = \begin{cases} \text{empty} & O_{GM}^{ge} < 0.5 \\ \text{unknown} & O_{GM}^{ge} = 0.5 \\ \text{occupied} & O_{GM}^{ge} > 0.5 \end{cases} \tag{4}$$

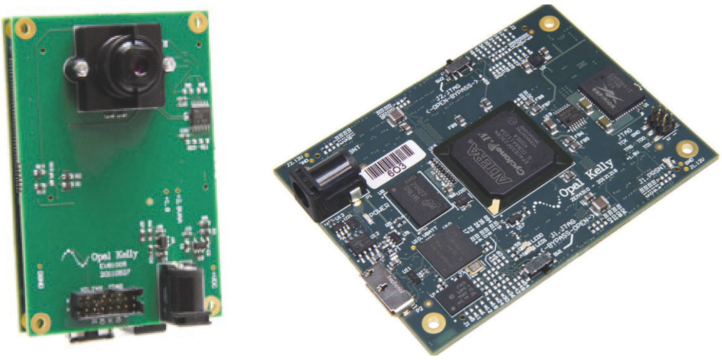
3.3.1. Image Binarization Module. In this module, an image coming from AIE module is segmented in two classes, which correspond to either floor (class 0) or obstacle regions (class 1). This is achieved by using a linear classifier trained with a gradient descent algorithm [43]. Here, each pixel is classified in one of the two classes according to its neighborhood (a 3 × 3 window) information; from the window, mean and standard deviation are also obtained and used by the classifier as feature vector.



(a)

(b)

Figure 3: Input image correction. (a) Original captured image. (b) Contrast enhanced image.



(a) Camera sensor module

(b) FPGA module

Figure 4: OpalKelly modules.

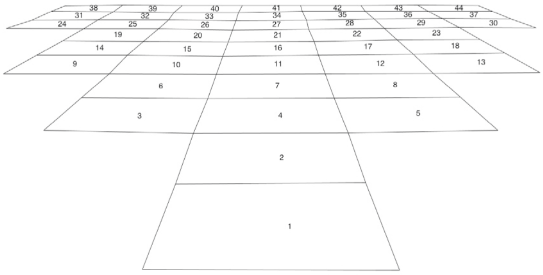


Figure 5: CC's output: grid template of chessboard-like calibration pattern.

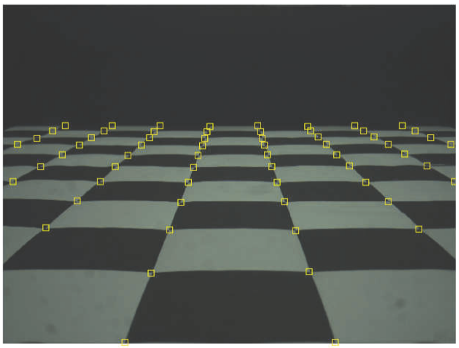


Figure 6: Output of Harris Corner Detector Algorithm. The detected corners are in yellow.

3.3.2. Odometry Module. Robot's navigation requires the position and orientation of the robot according to its local and global reference axe. This task is commonly achieved through the odometry estimation method, which is based on geometric equations that provide an estimation of robot's location by combining information obtained from wheel's encoders and actuators. This method is widely used because it only uses the robot's kinematic model and the external forces acting over the mechanism are not taken into account. In our proposal, the odometry module provides the robot's position and orientation when an image is captured or during navigation (see [38] for more details). This module provides input to the Probabilistic Occupancy Local Map and Update modules with respect to real-world's reference system $\{W\}$. The orientation of the system is provided to the Probabilistic Occupancy Local Map module, whereas the spatial position is given to the Update one.

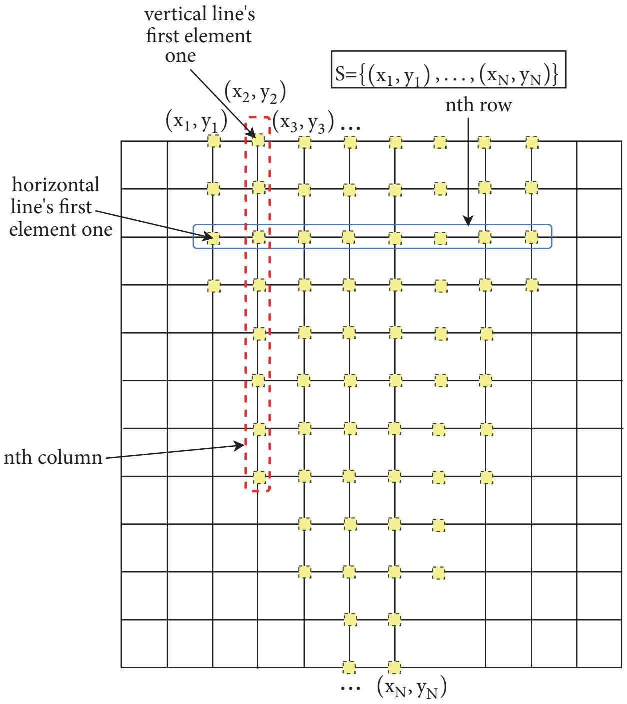

vertical line's first element one
(x_2, y_2)
$S=\{(x_1, y_1), \ldots, (x_N, y_N)\}$
nth row
(x_1, y_1)
$(x_3, y_3) \ldots$
horizontal line's first element one
nth column
$\ldots$ (x_N, y_N)

Figure 7: Representation of calibration chessboard-like pattern's components.

3.3.3. Probabilistic Occupancy Local Map Module. In this module, the Probabilistic Occupancy Local Maps (POLMs) are generated; each map has a dimension of 31 × 31 grid-elements (covering a surface of 3.1 × 3.1 m in the real world). To generate a probabilistic occupancy local map at time k, we take the current image given by Image Binarization module and the information of the grid generated in the CC block. For each grid-element in the binarized image, pixels that were classified as obstacles are counted and used together with the quantity of pixels in the grid-element (obtained from CC block) to calculate its occupancy probability $O^{ge}_{LM}(k) = objectPixels/pixelsGridElement$. A grid-element into the probabilistic occupancy local maps can have any of three occupancy states ($OS^{ge}_{LM}(k)$), which are given by

$$OS^{ge}_{LM}(k) = \begin{cases} \text{empty} & O^{ge}_{LM} < 0.5 \\ \text{unknown} & O^{ge}_{LM} = 0.5 \\ \text{occupied} & O^{ge}_{LM} > 0.5 \end{cases} \quad (5)$$

3.3.4. Update Module. This module updates the OGM by means of its current status and that of the POLMs. The update is carried out over each grid-element by comparing its current occupancy probability value $O^{ge}_{LM}(k)$ against its corresponding occupancy probability value $O^{ge}_{GM}(k-1)$ on the OGM in the previous time. Next, we present the resulting cases of such comparison for the current grid-element and (6) shows the update rule for each of them.

Case 1. $OS^{ge}_{GM}(k-1)$ is empty and $OS^{ge}_{LM}(k)$ is occupied or $OS^{ge}_{GM}(k-1)$ is occupied and $OS^{ge}_{LM}(K)$ is empty.

Case 2. $OS^{ge}_{LM}(k)$ is unknown and the value of $OS^{ge}_{GM}(k-1)$ is not taken into account:

$$O^{ge}_{GM}(k) = \begin{cases} P_R * O^{ge}_{LM}(k) + Ph * O^{ge}_{GM}(k-1) & \text{Case 1} \\ O^{ge}_{GM}(k-1) & \text{Case 2} \\ O^{ge}_{LM}(k) & \text{Else} \end{cases} \quad (6)$$

where $P_R = 0.25$ and $P_h = 0.75$ are weight factors.

In Table 1, cases and update rules for the Probabilistic Occupancy Global Map are summarized.

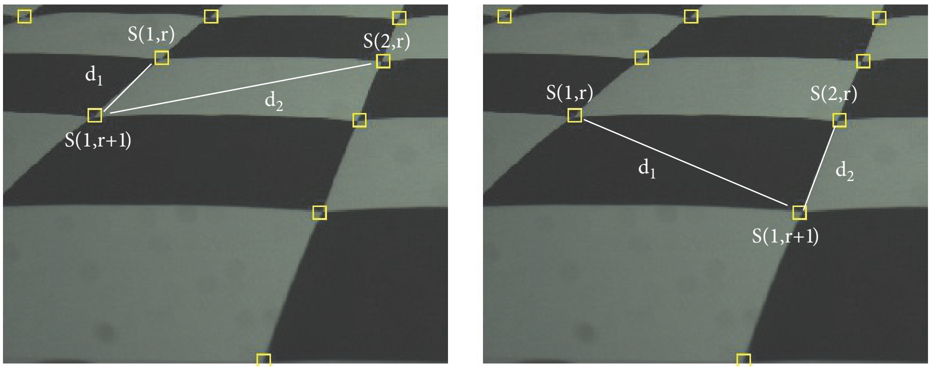
4. Experiments

To validate our approach, we carried out numerical tests in a controlled environment, i.e., a room of 3.8 × 8.5 m with fixed obstacles. A sketch of the environment is shown in Figure 11, where blocks in gray represent obstacles and some samples of captured images with different spatial location are shown; the arrows connected to sample images mark the direction or orientation of the robot's sight in the environment. Furthermore, the floor in the environment is covered with tiles of 50 × 50 cm, which were used as reference for comparing the real position of obstacles in the environment against the position of obstacles detected by our method. For capturing the environment's images the camera sensor was set over a robotic platform called NI LabVIEW robotics Starter Kit from the National Instrument Company (NI); it has two DC motors, two encoders, and a reconfigurable board SbRIO-9632 (Single Board Reconfigurable I/O). The utilized navigation method was the potential fields; basically, it calculates the attraction forces produced by the target's position and the repulsion forces associated to the objects; thus, the target's position attracts the robot; meanwhile the obstacle's forces repel it, and this allows the robot to navigate among obstacles and reach its target's position [38].

5. Results and Discussion

In Table 2, we present numerical results on the environment's metric information performed by the Camera Calibration process using a chessboard-like calibration pattern (see Figure 5). These values correspond to the grid's information per grid-element, and they were further used to estimate the occupancy probability of maps.

Results for the image segmentation module in the OGM construction process are shown in Figure 12. From there, we can observe the captured and resulting images in odd and even columns, respectively. In general, it can be observed that the module's output gives images with a good distinction between floor (black regions) and obstacle (white regions); for example, it can segment obstacles with significant size



(a) $S(1, r + 1)$ is in the same line that $S(1, r)$

(b) $S(1, r + 1)$ is in the same line that $S(2, r)$

Figure 8: Use of Euclidean distances to find the vertices of the grids.

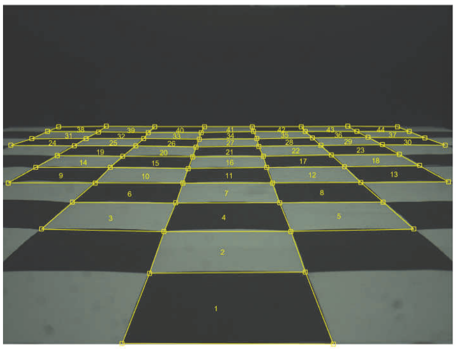


Figure 9: Calibration pattern with segmented grid-elements.

Table 2: Camera calibration results: number of pixels per grid-element.

No. Grid	No. Pixels	No. Grid	No. Pixels
1	107045	23	5166
2	47852	24	3382
3	26077	25	3587
4	27100	26	3608
5	26206	27	3706
6	15574	28	3660
7	15759	29	3659
8	15635	30	3553
9	10396	31	3021
10	10295	32	2714
11	10344	33	2874
12	10658	34	3165
13	10764	35	3090
14	6576	36	2781
15	7046	37	2793
16	7042	38	2033
17	6969	39	2356
18	6862	40	2486
19	5430	41	2288
20	5468	42	2027
21	5431	43	1974
22	5386	44	2034

such as the desktop' parts, but it is also capable of distinguishing small obstacles such as the power outlet.

Furthermore, in the even columns of Figure 13 we show the POLMs generated from the segmented images (odd columns). POLMs represent the visual field of the robot's camera as occupancy grids with three different colors, white, gray, and black, that represent the occupancy probability for high, unknown, and low, respectively. The unknown probability mainly refers to pixels either outside of the grid's region or behind an obstacle (outside of the visual field of the camera). Thus, for the POLMs shown in Figure 13 we have different grid-elements with different occupancy's level for the obstacles detected in the environment; for example, in Figure 13(b) we show the POLM corresponding to Figure 13(a) with left side grid-elements empty (black color) and some grid-elements near and at middle of the grid occupied (white color) detecting the desktop's part, and the rest of grid-elements, mostly at right side, have an unknown occupancy state because they are out of vision range (gray color).

Finally, in Figure 14 we present the output of our approach where a precise Occupancy Grip Map with 38 × 85 grid-elements has been constructed from the environment schematized in Figure 11. To get this, we run the system over the experimental environment following the trajectory shown in yellow in Figure 14. Also, we show some probabilities calculated by the system when an obstacle was detected.

To evaluate the performance of our proposal, we compare the quality of OGMs generated by our method against those obtained by Thrun's updating rule [39], both applied to the images shown in previous section. Instead of performing a visual inspection of generated OGMs, which becomes highly subjective, we use OGM evaluation metrics which provide quantitative values and allow a numeric comparison.

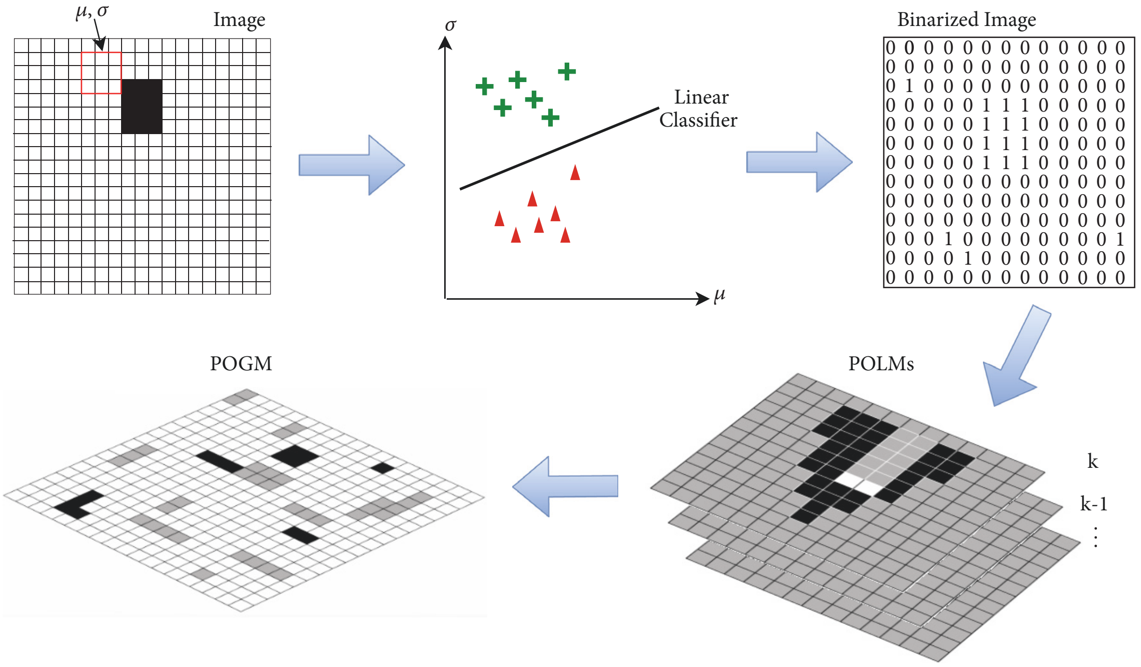


Figure 10: POGM generation process: POLMs are obtained from the captured images in order to build the POGM.

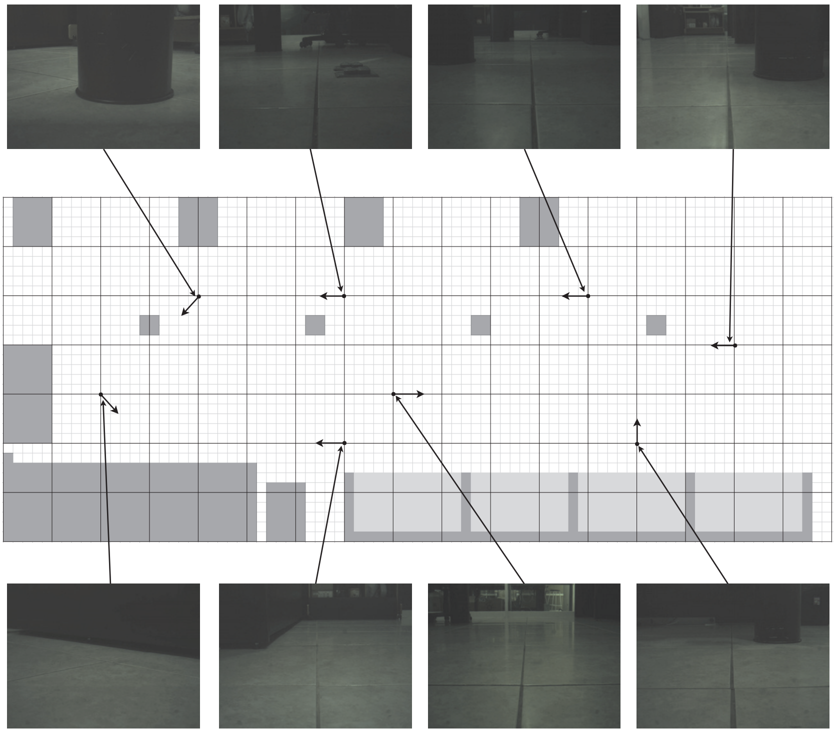


Figure 11: Sketch of the experimental environment.

Several OGM evaluation metrics have been proposed in the state of the art (e.g., [19, 26, 44–46]). We use three of them: Map score [26], Precision, and Recall [19], which are briefly described as follows.

(i) Map score metric compares the M and N maps, through the sum of squared differences between their grid-elements $m_{x,y} \in M$ and $n_{x,y} \in N$, respectively, at position (x, y). Generally, the $MapScore$ (≥ 0) is used

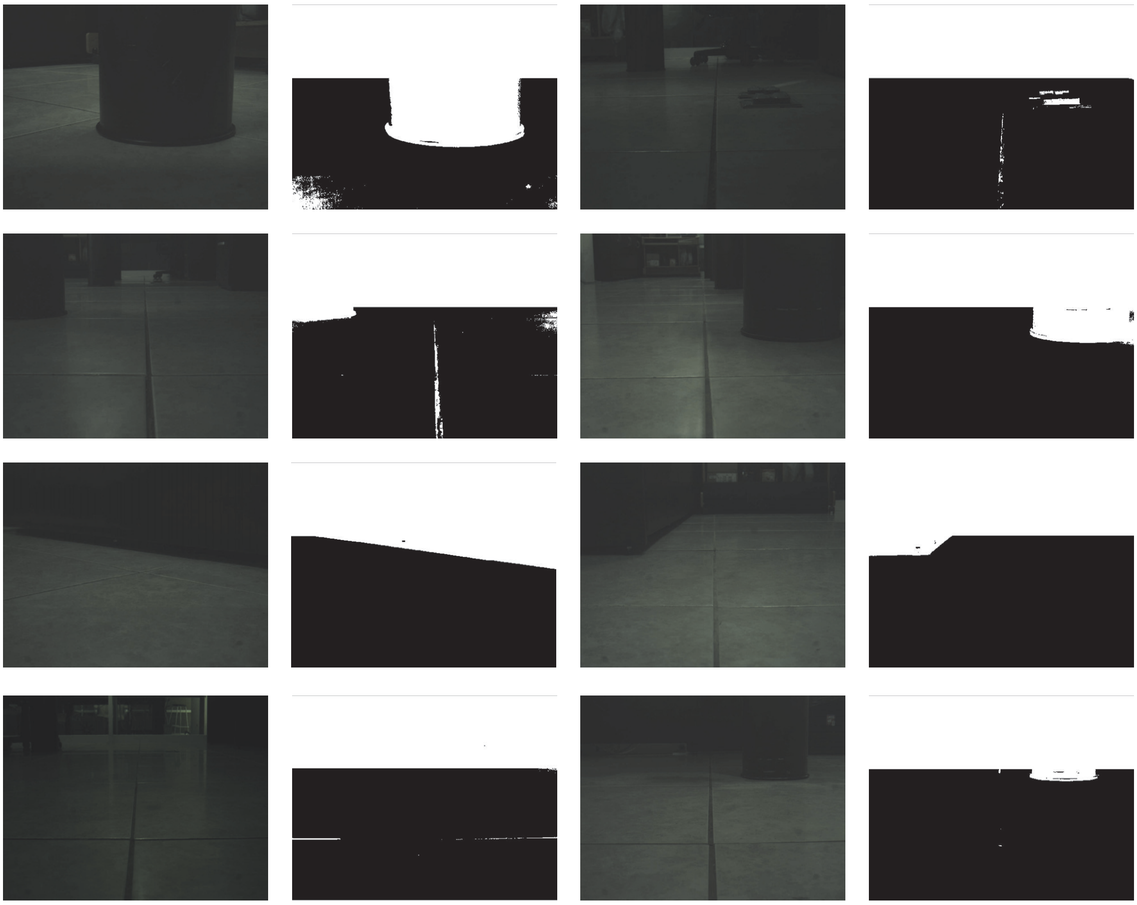


Figure 12: Binarization and segmentation (even columns) of captured images (odd columns), where black and white regions correspond to floor and obstacles, respectively.

to compare an ideal map of the environment and a map generated by means of a method; thus, the lower the *MapScore* is, the more alike maps are. Next, the metric is shown:

$$MapScore = \sum_{m_{x,y}\in M, n_{x,y}\in N} \left(m_{x,y} n_{x,y}\right)^2 \quad (7)$$

(ii) Precision and Recall metrics evaluate the quality of generated maps through a precision recovery analysis. The generated maps are treated as classification problems in which the objective is to correctly classify the pixels into occupied or empty classes. The pixels occupied in the reference environment are considered as true positives and the empty pixels as true negatives. The pixels of any other detected object are defined as false positive. Next, these metrics are shown:

$$Precision\ (PRE) = \frac{True\ Positives}{Detected\ Occupied\ Space} * 100 \quad (8)$$

T he Precision metric represents the percentage of obstacles correctly detected:

$$Recall\ (REC) = \frac{True\ Positives}{Actual\ Occupied\ Space} * 100 \quad (9)$$

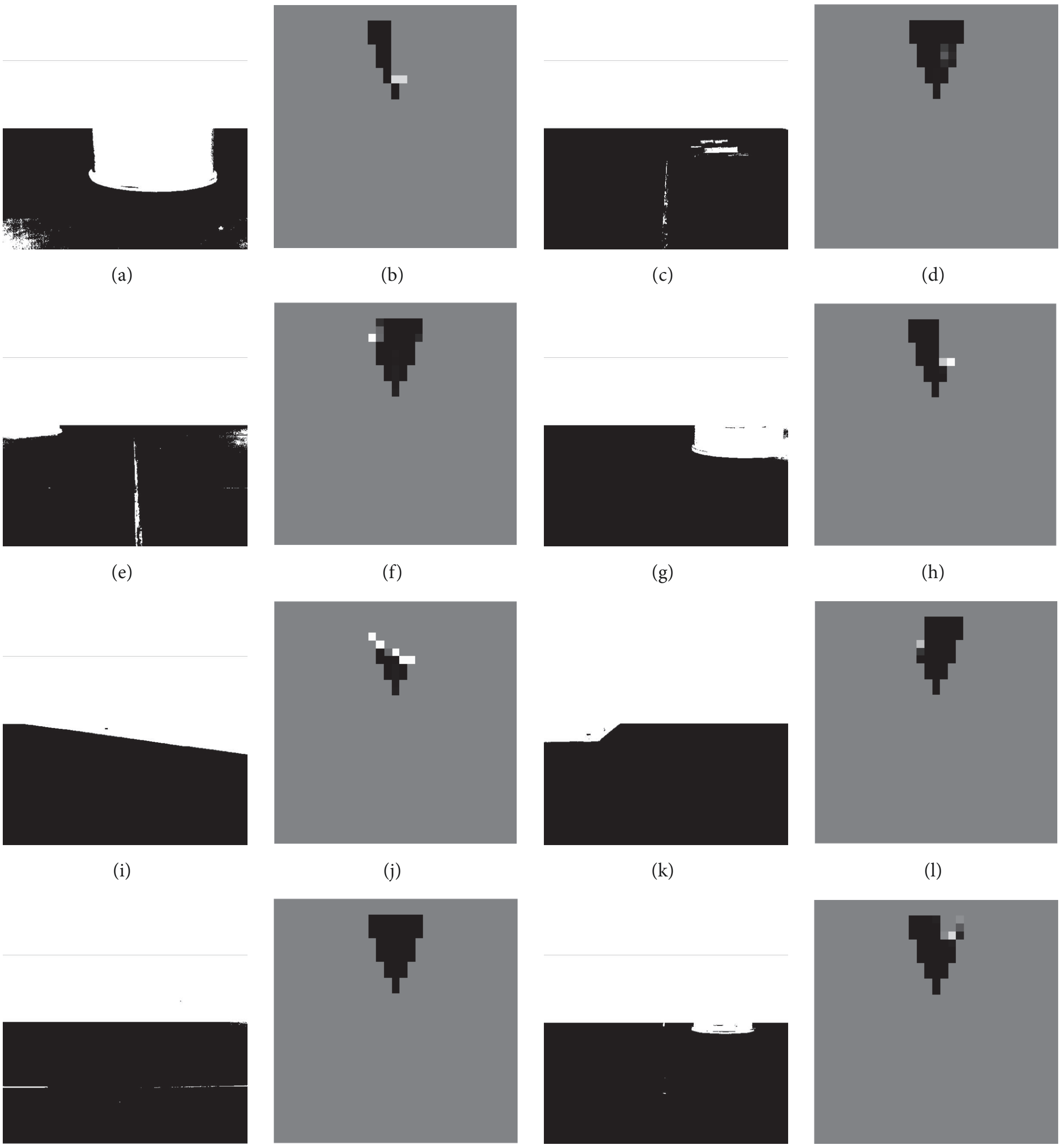
And the Recall metric represents the percentage of obstacles correctly detected among all obstacles in the reference environment.

In order to use the aforementioned OGM evaluation metrics and compare our method against Thrun's method, we obtained the ideal map and generated the OGMs by using our proposal and Thrun's method; these are shown in Figures 15(a), 15(b), and 15(c), respectively.

The results of the OGM evaluation metrics applied to the generated OGMs are shown in Table 3. It can be numerically observed that our proposed method gets lightly better results for the three OGM evaluation metrics used for comparison. Even when numeric values are close for both generated OGMs, it can be argued that our proposal is lightly more alike to the ideal map because it has a lower value for the Score Map metric and has obstacles correctly detected according to Precision and Recall metrics because it has greater values than those obtained by Thrun's method.

6. Conclusions and Future Work

In this work, we have successfully contributed to the field of visual robot navigation by improving the main process associated with the construction of navigation maps using



(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k) (l)

(m) (n) (o) (p)

Figure 13: Probabilistic Occupancy Local Maps for binarized and segmented images.

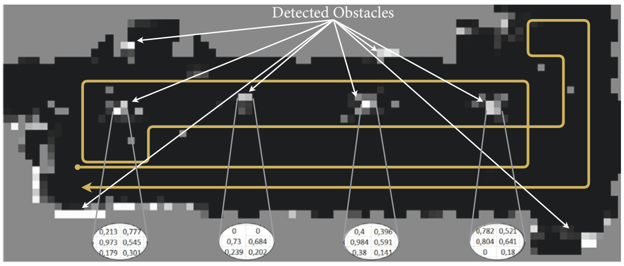


Figure 14: The final Occupancy Grid Map of the environment used for experimentation.

Table 3: Values obtained from OGM evaluation metrics.

	Map Score	PRE	REC
Proposed Method	206.7	95.49	79.94
Thrun's Method	214.6	95.19	79.90

occupancy grids. It is well known that OGM has been the predominant algorithm for navigation tasks in mobile robots such as localization, path planning, collision avoidance, people finding, etc.; however it is also true that this method demands high computational cost, specially for large map construction. Thus, we have proposed a novelty method

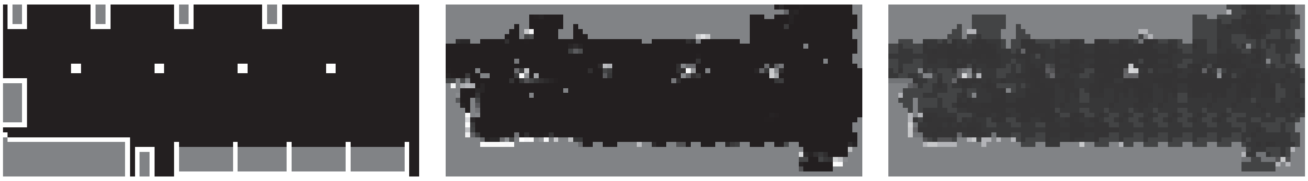

(a) Ideal Map (b) Our method (c) Thrun's method

Figure 15: OGMs generated for comparison.

that uses fewer homography operations, which significantly reduces the amount of calculations compared to previous methods that require camera calibration; this is achieved by keeping the extrinsic camera parameters fixed. Furthermore, this improvement of the CC process allowed us to treat simultaneously the two main processes involved in OGM-based visual navigation systems, i.e., camera calibration and maps construction. By treating simultaneously these processes, we get more precise navigation maps because the errors generated by rounding values while building maps are avoided. Three OGM metrics were used to compare our proposal against the Thrun's method, and both generated maps that are quite similar but the maps generated by our proposal obtained values which indicate that they and the ideal map are more alike. A final remark is that our method is able to truly detect small objects as has been shown in the results.

On the other hand, OGM-based methods depend also on a correct positioning of the robot on the map, just based on the measurement of the odometry, which can generate a large cumulative error during a continuous operation of the algorithm. This problem is intended to be solved in a future work using visual odometry to complement the measurements, either by analyzing the optical flow or by using markers and in this way the displacement error is estimated and corrected. Also, to improve the processing time, a FPGA-based architecture can be implemented to take advantage of the parallelism features of our method and thus accelerate the current implementation.

Data Availability

The images used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This reserch has been supported by the CONACYT project FC2016-1961 "Neurociencia Computacional: de la teoría al desarrollo de sistemas neuromórficos," Publicación Financiada con recursos PFCE 2019.

References

[1] K. Kant and S. W. Zucker, "Planning collision-free trajectories in time-varying environments: a two-level hierarchy," *The Visual Computer*, vol. 3, no. 5, pp. 304–313, 1988.

[2] S. Thrun, "Learning occupancy grids with forward models," in *Proceedings of the RSJ/IEEE International Conference on Intelligent Robots and Systems (IROS '01)*, pp. 1676–1681, Wailea, HI, USA, 2001.

[3] H. Moravec and A. Elfes, "High resolution maps from wide angle sonar," in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 2, pp. 116–121, March 1985.

[4] R. Jiang, S. Yang, S. S. Ge, H. Wang, and T. H. Lee, "Geometric map-assisted localization for mobile robots based on uniform-gaussian distribution," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 789–795, 2017.

[5] R. Drouilly, P. Rives, and B. Morisset, "Fast hybrid relocation in large scale metric-topologic-semantic map," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '14)*, pp. 1839–1845, September 2014.

[6] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *The Computer Journal*, vol. 22, no. 6, pp. 46–57, 1989.

[7] E. Horvath and C. R. Pozna, "Probabilistic occupancy grid map building for Neobotix MP500 robot," in *Proceedings of the 13th Workshop on Positioning, Navigation and Communication (WPNC '16)*, pp. 1–4, October 2016.

[8] B. J. Julian, S. Karaman, and D. Rus, "On mutual information-based control of range sensing robots for mapping applications," *International Journal of Robotics Research*, vol. 33, no. 10, pp. 1375–1392, 2014.

[9] G. Mingas, E. Tsardoulias, and L. Petrou, "An FPGA implementation of the SMG-SLAM algorithm," *Microprocessors and Microsystems*, vol. 36, no. 3, pp. 190–204, 2012.

[10] T. Rakotovao, J. Mottin, D. Puschini, and C. Laugier, "Multi-sensor fusion of occupancy grids based on integer arithmetic," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '16)*, pp. 1854–1859, May 2016.

[11] J. Moras, V. Cherfaoui, and P. Bonnifait, "Credibilist occupancy grids for vehicle perception in dynamic environments," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '11)*, pp. 84–89, May 2011.

[12] D. Pagac, E. M. Nebot, and H. Durrant-Whyte, "An evidential approach to map-building for autonomous vehicles," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 4, pp. 623–629, 1998.

[13] R. Dube, M. Hahn, M. Schutz, J. Dickmann, and D. Gingras, "Detection of parked vehicles from a radar based occupancy grid," in *Proceedings of the 25th IEEE Intelligent Vehicles Symposium (IV '14)*, pp. 1415–1420, June 2014.

[14] D. A. De Lima, A. C. Victorino, and A. De Miranda Neto, “A 2D/3D environment perception approach applied to sensor-based navigation of automated driving systems,” in *Proceedings of the 14th Latin American Robotics Symposium and 5th Brazilian Symposium on Robotics (LARS-SBR ’17)*, pp. 1–6, IEEE, November 2017.

[15] C. Yu, V. Cherfaoui, and P. Bonnifait, “Evidential occupancy grid mapping with stereo-vision,” in *Proceedings of the IEEE Intelligent Vehicles Symposium (IV ’15)*, pp. 712–717, July 2015.

[16] R. Danescu, F. Oniga, and S. Nedevschi, “Modeling and tracking the driving environment with a particle-based occupancy grid,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1331–1342, 2011.

[17] M. Perrollaz, A. Spalanzani, and D. Aubert, “Probabilistic representation of the uncertainty of stereo-vision and application to obstacle detection,” in *Proceedings of the IEEE Intelligent Vehicles Symposium (IV ’10)*, pp. 313–318, June 2010.

[18] M. Perrollaz, J.-D. Yoder, and C. Laugier, “Using obstacles and road pixels in the disparity-space computation of stereo-vision based occupancy grids,” in *Proceedings of the 13th International IEEE Conference on Intelligent Transportation Systems (ITSC ’10)*, pp. 1147–1152, September 2010.

[19] C. Georgiou, S. Anderson, and T. Dodd, “Constructing informative Bayesian map priors: A multi-objective optimisation approach applied to indoor occupancy grid mapping,” *International Journal of Robotics Research*, vol. 36, no. 3, pp. 274–291, 2017.

[20] A. S. Huang, A. Bachrach, P. Henry et al., “Visual odometry and mapping for autonomous flight using an RGB-D camera,” *Springer Tracts in Advanced Robotics*, vol. 100, pp. 235–252, 2017.

[21] E. Kaufman, T. Lee, and Z. Ai, “Autonomous exploration by expected information gain from probabilistic occupancy grid mapping,” in *Proceedings of the IEEE International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR ’16)*, pp. 246–251, December 2016.

[22] S.-I. Oh and H.-B. Kang, “Fast occupancy grid filtering using grid cell clusters from LIDAR and stereo vision sensor data,” *IEEE Sensors Journal*, vol. 16, no. 19, pp. 7258–7266, 2016.

[23] T. Yuan, D. S. Nuss, G. Krehl, M. Maile, and A. Gern, “Fundamental properties of dynamic occupancy grid systems for vehicle environment perception,” in *Proceedings of the 6th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP ’15)*, pp. 153–156, December 2015.

[24] F. Homm, N. Kaempchen, J. Ota, and D. Burschka, “Efficient occupancy grid computation on the GPU with lidar and radar for road boundary detection,” in *Proceedings of the IEEE Intelligent Vehicles Symposium (IV ’10)*, pp. 1006–1013, June 2010.

[25] O. Sang and K. Hand-Bong, “Fast occupancy grid filtering using grid cell clusters from LIDAR and stereo vision sensor data,” *IEEE Sensor Journal*, vol. 16, no. 19, pp. 7258–7266, 2016.

[26] T. Colleens and J. J. Colleens, “Occupancy grid mapping: An empirical evaluation,” in *Proceedings of the Mediterranean Conference on Control & Automation (MED ’07)*, pp. 1–6, 2007.

[27] Z. Zhang, “A flexible new technique for camera calibration,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.

[28] Y. Dong, X. Ye, and X. He, “A novel camera calibration method combined with calibration toolbox and genetic algorithm,” in *Proceedings of the 11th IEEE Conference on Industrial Electronics and Applications (ICIEA ’16)*, pp. 1416–1420, June 2016.

[29] S. J. Bezuszka, M. Kenney, and L. Silvey, “Tessellations: the geometry of patterns,” in *Creative Publications Math Design. Creative Publications*, 1977.

[30] J. Salvi, X. Armangué, and J. Batlle, “A comparative review of camera calibrating methods with accuracy evaluation,” *Pattern Recognition*, vol. 35, no. 7, pp. 1617–1635, 2002.

[31] F. Remondino and C. Fraser, “Digital camera calibration methods: Considerations and comparisons,” in *Proceedings of the The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 36, pp. 266–272, Dresden, Germany, 2006.

[32] S. El Hazzat, M. Merras, N. El Akkad, A. Saaidi, and K. Satori, “3D reconstruction system based on incremental structure from motion using a camera with varying parameters,” *The Visual Computer*, vol. 34, no. 10, pp. 1443–1460, 2018.

[33] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2nd edition, 2003.

[34] G. E. Zaikov, *Analysis and Performance of Engineering Materials: Key Research and Development*, Apple Academic Press, 2015.

[35] M. Yguel, O. Aycard, and C. Laugier, “Efficient GPU-based construction of occupancy grids using several laser range-finders,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS ’06)*, pp. 105–110, October 2006.

[36] S. Liu, W. Cui, Y. Wu, and M. Liu, “A survey on information visualization: recent advances and challenges,” *The Visual Computer*, vol. 30, no. 12, pp. 1373–1393, 2014.

[37] L. Matthies and A. Elfes, “Integration of sonar and stereo range data using a grid-based representation,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 727–733, 1988.

[38] D.-L. Almanza-Ojeda, Y. Gomar-Vera, and M.-A. Ibarra-Manzano, “Occupancy map construction for indoor robot navigation,” in *Robot Control*, InTech, 2016.

[39] S. Thrun, “Learning metric-topological maps for indoor mobile robot navigation,” *Artificial Intelligence*, vol. 99, no. 1, pp. 21–71, 1998.

[40] B. Jia, R. Liu, and M. Zhu, “Real-time obstacle detection with motion features using monocular vision,” *The Visual Computer*, vol. 31, no. 3, pp. 281–293, 2015.

[41] H. Farid, “Blind inverse gamma correction,” *IEEE Transactions on Image Processing*, vol. 10, no. 10, pp. 1428–1433, 2001.

[42] C. Harris and M. Stephens, “A combined corner and edge detector,” in *Proceedings of the 4th Alvey Vision Conference*, vol. 15, pp. 10–5244, Manchester, UK, 1988.

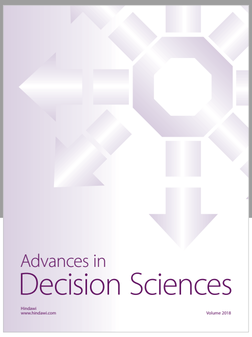
[43] D. W. Patterson, *Artificial Neural Networks*, Prentice-Hall, 1996.

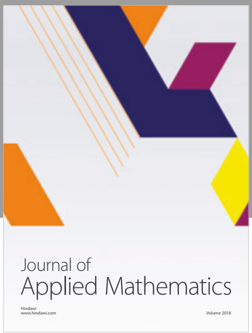
[44] D. Cho, “Certainty grid representation for robot navigation by a bayesian method,” *Robotica*, vol. 8, no. 2, pp. 159–165, 1990.

[45] A. Elfes, “Dynamic control of robot perception using stochastic spatial models,” in *Information Processing in Mobile Robots*, Springer-Verlag, Berlin, Germany, 1991.

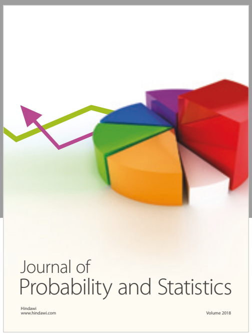
[46] C. Martin and P. Hans Moravec, “Robot evidence grids,” Technical Report, Carnegie-Mellon Univ Pittsburgh Pa Robotics Inst, 1996.

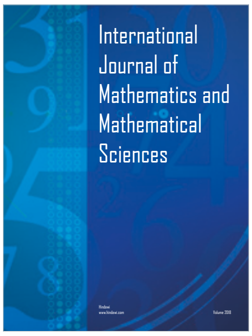
Advances in
Operations Research


Advances in
Decision Sciences

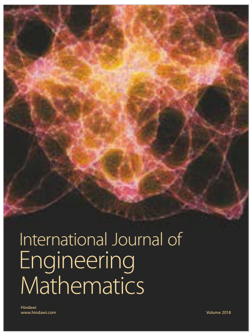

Journal of
Applied Mathematics

The Scientific
World Journal

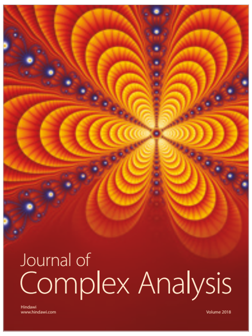

Journal of
Probability and Statistics

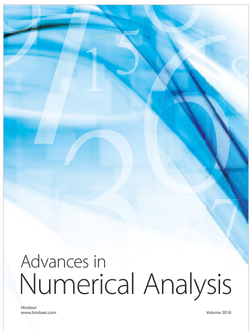

International
Journal of
Mathematics and
Mathematical
Sciences

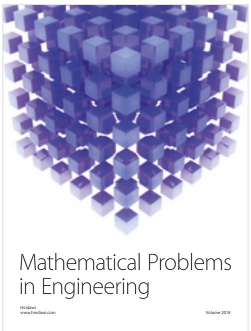
Journal of
Optimization

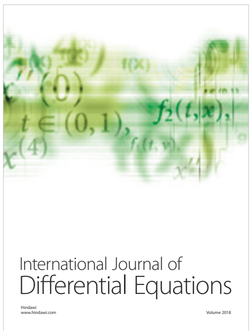

International Journal of
Engineering
Mathematics

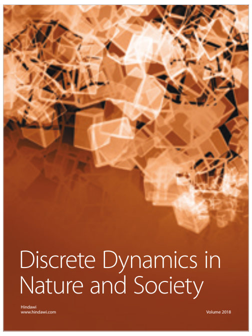
International Journal of
Analysis

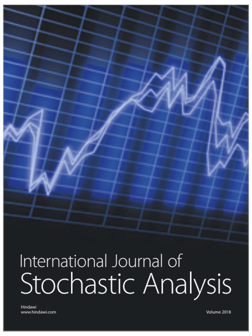

Journal of
Complex Analysis


Advances in
Numerical Analysis


Mathematical Problems
in Engineering

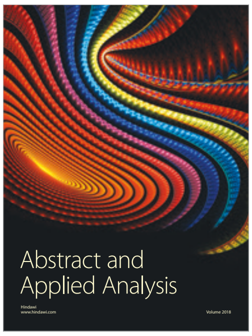

International Journal of
Differential Equations

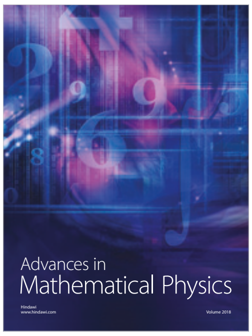

Discrete Dynamics in
Nature and Society


International Journal of
Stochastic Analysis

Journal of
Mathematics

Journal of
Function Spaces


Abstract and
Applied Analysis


Advances in
Mathematical Physics