An incremental hybrid approach to indoor modeling

Marco A. Gutiérrez Pilar Bachiller Luis J. Manso Pablo Bustos Pedro Núñez
Department of Computer and Communication Technology, University of Extremadura, Spain

***Abstract*— Most of mobile robots basic functions are highly dependent on a model of their environment. Proper modeling is crucial for tasks such as local navigation, localization or route planning. This paper describes a novel solution for building models of indoor environments. We use a 3D Laser on a mobile robot to scan the surroundings and obtain sensing information. While the robot explores the environment, perceived points are clustered forming models of rooms. Room modeling is solved using a new variation of the Hough transform. The result of the modeling process is a topological graph that represents the rooms (nodes) and their connecting doors (edges). Each node in this representation contains the metric parametrization of the room model. Using this basic metric information, robots do not need to maintain in parallel a metric map of the environment. Instead, this metric map can be totally or partially built from the topological representation whenever it is necessary. To test the approach we have carried out a modeling experiment of a real environment, obtaining promising results.**

***Index Terms*— Environment modeling, hybrid representation, active exploration, 3D laser scanning.**

I. Introduction

In recent years, the problem of map building has become an important research topic in the field of robotics. A wide variety of proposals use dense metric maps to represent the robot environment. While a metric representation is necessary for solving certain robot tasks, many others require a more qualitative organization of the environment. Topological maps provide such a qualitative description of the space and constitute a good support to the metric information.

Several approaches on mobile robotics propose the use of topological representation to complement the metric information of the environment. In [16] it is proposed to create off-line topological graphs by partitioning metric maps into regions separated by narrow passages. In [14] the environment is represented by a hybrid topological-metric map composed of a set of local metric maps called *islands of reliability*. [17] describes the environment using a global topological map that associates places which are metrically represented by infinite lines belonging to the same places. [18] constructs a topological representation as a route graph using Voronoï diagrams. In [19] the environment is represented by a graph whose nodes are crossings (corners or intersections). [9] organizes the information of the environment in a graph of planar regions. [3] proposes an off-line method that builds a topological representation, whose nodes correspond to rooms, from a previously obtained metric map.

In this paper, we propose a novel incremental modeling method that provides a hybrid topological/metric representation of indoor environments. Our approach improves the current state of the art in several aspects. Firstly, as in [3], the topological space is represented by a graph whose nodes encode rooms and whose edges describe connections between rooms. However, in our proposal, the topological representation is incrementally built from the local information provided by the sensor. This means that no global metric map is needed to extract a topological description of the environment. In addition, instead of maintaining in parallel a dense metric map, each topological node contains a minimal set of metric information that allows building a map of a part of the environment when it is needed. This approach reduces drastically the computation the robot must perform to maintain an internal representation of its surroundings. In addition, it can be very helpful for solving certain tasks in an efficient way, such as global navigation or self-localization.

We also present a new variation of the Hough transform used for room modeling. Under the rectangularity assumption, this method provides the geometrical parametrization of a room from a set of points. It is shown how this proposal improves the results compared to other approaches.

The rest of the paper is organized as follows. In section II, an overview of the approach is presented. Section III describes the modeling method that provides a description of the environment in terms of rooms and their connections. Along section IV, the process for creating the proposed hybrid representation is detailed. Experimental results in a real environment are presented in section V. Finally, section VI summarizes the main conclusions of our work.

II. Overview of the approach

As a first approach towards environment modeling, we focus on indoor environments composed by several rooms connected through doors. Rooms are considered approximately rectangular and contain several objects on the floor.

To solve the low-level perceptual process, we have equipped one of our mobile robots with a motorized 2D LRF (Laser Range Finder). Stereo cameras, static 2D LRFs and 3D LRFs constitute an alternative for this purpose. It is possible nowadays to use stereo vision or even the very popular RGB-D primesense sensor[1] to retrieve depth from images and, in consequence, be able to map the robot surroundings. However most of these vision studied techniques are performed under almost ideal circumstances. Uniformly colored surfaces or light variations are some of the problems these solutions might face. In addition, compared to LRF performance, sensors like RGB-D get small fields of view and low depth precision (3cm in 3m scan). LRFs constitute a strong alternative to these sensors, especially when facing not ideal environments. A wide variety of this kind of sensors have became lately available: point range finders, 2D LRFs and 3D LRFs. 3D LRFs seem promising, but their high cost makes them in practice less usable than other sensors. Motorized 2D LRFs have arisen as a good solution combining important advantages in relation to other sensors for this applications.

To cover the whole 3D spectrum, the 2D LRF is attached to a step motor that makes it scan the whole hemisphere in front of the mobile robot. The resolution is made to be directly dependent on the scanning speed and the robot is able to adjust the related parameters accordingly to its needs. During the exploration, perceived points are stored in a 3D occupancy grid that constitutes a discrete representation of a particular zone of the environment. This occupancy grid is locally used, so, when the robot gets into a new room, the grid is reseted. Each cell of this grid contains the certainty degree about the occupancy of the corresponding volume of the environment. The certainty increases if a point is perceived over time in the same position and decreases when the point vanishes from the space covered by the sensor. Thus, stable points produce higher occupancy values than unstable ones. Cells with a high certainty degree are used for detecting a room model fitting the set of perceived points. Once the model of the current room can be considered stable, it is stored in an internal representation that maintains topological and metric information of the environment.

In this representation, the environment is described as an undirected graph whose vertices represent the different explored rooms (see figure 1). An edge linking two vertices expresses the existence of a door that connects two rooms. This is a very useful representation for the robot to effectively move around man-made environments. For instance, the robot could analyze the graph to obtain the minimum path connecting any two rooms. Moreover, this representation can be extended using recursive descriptions to express more complex world structures like buildings. Thus, a building could be represented by a node containing several interconnected subgraphs. Each subgraph would represent a floor of the building and contain a description of the interconnections between the different rooms and corridors in it.

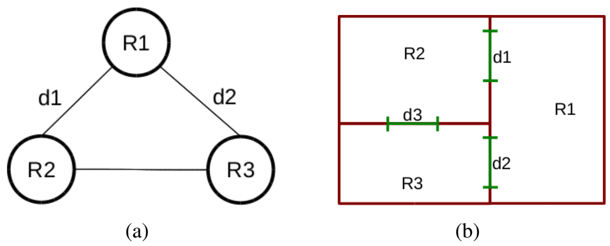


Fig. 1. Topological representation (a) of an environment (b) composed by three intercommunicated rooms.

Using this topological graph, a minimal set of metric information is maintained. Specifically, each vertex of the graph stores the parametric description of the corresponding room and its doors. Using this basic metric information, the robot does not need to maintain in parallel a metric map of the environment. Instead, a total or partial metric representation can be recovered whenever it is necessary from the topological one.

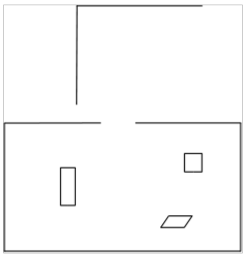
To deal with the uncertainty derived from odometric and sensor errors, we follow a non simultaneous approach to localization and modeling. Modeling errors are minimized by performing specific actions directed to quickly obtain measures around the robot. Once a model is fixed, errors in the following models, i.e. adjacent rooms, are canceled by applying geometric restrictions. Finally, detected loop closings are used through a global minimization procedure to further reduce modeling errors. Localization errors are canceled by continuously rebuilding a new model that is compared to the current one. Using an estimated rectangular model instead of the raw measures is an effective procedure to update the position of the robot relative to the current model, since it is less sensitive to wrong measurements. As long as the rectangular hypothesis is coherent with the real world around the robot, these methods work robustly in real situations. More details on this are given on section IV.

III. Rooms and doors modeling

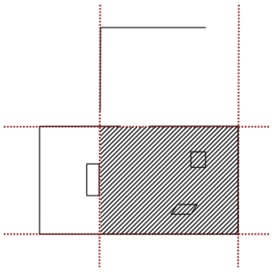
Since rooms are assumed to be rectangular and its walls perpendicular to the floor, the problem of modeling a room from a set of points can be treated as a rectangle detection problem using the projections of those points onto the floor plane. Several rectangle detection techniques can be found in the literature [5, 6, 15]. Most of them are based on a search in the 2D point space (for instance, a search in the edge representation of an image) using line primitives. These methods are computationally expensive and can be very sensitive to noisy data. In order to solve the modeling problem in an efficient way, we propose a new rectangle detection technique based on a search in the parameter space using a variation of the Hough Transform [13, 2].

For line detection, several variations of the Hough Transform have been proposed [8, 11]. The extension of the Hough Transform for rectangle detection is not new. [20] proposes a *Rectangular Hough Transform* used to detect the center and orientation of a rectangle with known dimensions. [4] proposes a *Windowed Hough Transform* that consists of searching rectangle patterns in the Hough space of every window of suitable dimensions of an image.

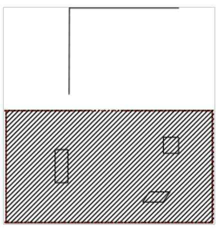
Our approach for rectangle detection uses a 3D version of the Hough Transform that facilitates the detection of segments instead of lines. This allows considering only those points that belong to the contour of a rectangle in the detection process, which is very important to obtain reliable results. For instance, consider the 2D view of the occupancy grid that is shown in figure 2(a). In this situation, the robot has perceived all the walls and objects of the room it is located and, partially, two walls of the adjoining room. Figures 2(b) and 2(c) show the results of the rectangle detection process using variations of the Hough transform based on lines and segments, respectively. The lined region of both figures corresponds to the detected rectangle. As it can be observed, a method based on lines, as the one proposed in [4], considers points that can be situated outside the rectangle, leading sometimes to wrong results. Nevertheless, the proposed variation of the Hough transform takes into account only those points belonging to the four segments of the rectangle providing always the best rectangle pattern.



(a) 2D view of the occupancy grid.



(b) Rectangle detection using lines.



(c) Rectangle detection using segments.

Fig. 2. Rectangle detection using the variation of the Hough transform based on lines (b) and the proposed one based on segments (c).

A. Room detection

In our room detection method, the Hough space is parameterized by (θ, d, p), being θ and d the parameters of the line representation ($d = x\cos(\theta) + y\sin(\theta)$) and $|p|$ the length of a segment in the line. For computing p it is assumed that one of the extreme points of its associated segment is initially fixed and situated at a distance of 0 to the perpendicular line passing through the origin. Under this assumption, being (x, y) the other extreme point of the segment, its *signed* length p can be computed as:

$$p = x\cos(\theta + \pi/2) + y\sin(\theta + \pi/2) \tag{1}$$

Using this representation, any point (x, y) contributes to those points (θ, d, p) in the Hough space that verifies:

$$d = x\cos(\theta) + y\sin(\theta) \tag{2}$$

and

$$p >= x\cos(\theta + \pi/2) + y\sin(\theta + \pi/2) \tag{3}$$

Equation 2 represents every line intersecting the point as in the original Hough Transform. The additional condition expressed by equation 3 limits the point contribution to those line segments containing the point. This allows computing the total number of points included in a given segment. For instance, given a segment with extreme points $V_i = (x_i, y_i)$ and $V_j = (x_j, y_j)$ and being H the 3D Hough space, the number of points that belongs to the segment, which is denoted as $H_{i\leftrightarrow j}$, can be computed as:

$$H_{i\leftrightarrow j} = |H(\theta_{i\leftrightarrow j}, d_{i\leftrightarrow j}, p_i) - H(\theta_{i\leftrightarrow j}, d_{i\leftrightarrow j}, p_j)| \tag{4}$$

where $\theta_{i\leftrightarrow j}$ and $d_{i\leftrightarrow j}$ are the parameters of the common line to both points and p_i and p_j are the *signed* lengths of the two segments with non-fixed extreme points V_i and V_j, respectively, according to equation 1.

Since a rectangle is composed of four segments, the 3D Hough space parameterized by (θ, d, p) allows computing the total number of points included in the contour of the rectangle. Thus, considering a rectangle expressed by its four vertices $V_1 = (x_1, y_1)$, $V_2 = (x_2, y_2)$, $V_3 = (x_3, y_3)$ and $V_4 = (x_4, y_4)$ (in clockwise order), the number of points of its contour, denoted as H_r, can be computed as:

$$H_r = H_{1\leftrightarrow 2} + H_{2\leftrightarrow 3} + H_{3\leftrightarrow 4} + H_{4\leftrightarrow 1} \tag{5}$$

Considering the restrictions about the segments of the rectangle and using the equation 4, each $H_{i\leftrightarrow j}$ of the expression 5 can be rewritten as follows:

$$H_{1\leftrightarrow 2} = |H(\alpha, d_{1\leftrightarrow 2}, d_{4\leftrightarrow 1}) - H(\alpha, d_{1\leftrightarrow 2}, d_{2\leftrightarrow 3})| \tag{6}$$

$$H_{2\leftrightarrow 3} = |H(\alpha+\pi/2, d_{2\leftrightarrow 3}, d_{1\leftrightarrow 2}) - H(\alpha+\pi/2, d_{2\leftrightarrow 3}, d_{3\leftrightarrow 4})| \tag{7}$$

$$H_{3\leftrightarrow 4} = |H(\alpha, d_{3\leftrightarrow 4}, d_{2\leftrightarrow 3}) - H(\alpha, d_{3\leftrightarrow 4}, d_{4\leftrightarrow 1})| \tag{8}$$

$$H_{4\leftrightarrow 1} = |H(\alpha+\pi/2, d_{4\leftrightarrow 1}, d_{3\leftrightarrow 4}) - H(\alpha+\pi/2, d_{4\leftrightarrow 1}, d_{1\leftrightarrow 2})| \tag{9}$$

being α the orientation of the rectangle and $d_{i\leftrightarrow j}$ the normal distance from the origin to the straight line defined by the points V_i and V_j.

Since H_r expresses the number of points in a rectangle r defined by $(\alpha, d_{1\leftrightarrow 2}, d_{2\leftrightarrow 3}, d_{3\leftrightarrow 4}, d_{4\leftrightarrow 1})$, the problem of obtaining the best rectangle given a set of points can be solved by finding the combination of $(\alpha, d_{1\leftrightarrow 2}, d_{2\leftrightarrow 3}, d_{3\leftrightarrow 4}, d_{4\leftrightarrow 1})$ that maximizes H_r. This parametrization of the rectangle can be transformed into a more practical representation defined by the five-tuple (α, x_c, y_c, w, h), being (x_c, y_c) the central point of the rectangle and w and h its dimensions. This transformation can be achieved using the following expressions:

$$x_c = \frac{d_{1\leftrightarrow 2} + d_{3\leftrightarrow 4}}{2}\cos(\alpha) - \frac{d_{2\leftrightarrow 3} + d_{4\leftrightarrow 1}}{2}\sin(\alpha) \tag{10}$$

$$y_c = \frac{d_{1\leftrightarrow 2} + d_{3\leftrightarrow 4}}{2}\sin(\alpha) + \frac{d_{2\leftrightarrow 3} + d_{4\leftrightarrow 1}}{2}\cos(\alpha) \tag{11}$$

$$w = d_{2\leftrightarrow 3} - d_{4\leftrightarrow 1} \tag{12}$$

$$h = d_{3\leftrightarrow 4} - d_{1\leftrightarrow 2} \tag{13}$$

In order to compute H_r, the parameter space H is discretized assuming the rank $[-\pi/2, \pi/2]$ for θ and $[d_{min}, d_{max}]$ for d and p, being d_{min} and d_{max} the minimum and maximum distance, respectively, between a line and the origin. Taking some sample step for each parameter and being G the 3D occupancy grid and τ the minimum occupancy value to consider a non-empty region of the environment, the proposed method for room modeling can be summarized in the following steps:

1) Initialize all the cells of the discrete Hough space H to 0.
2) For each cell, $G(x_d, y_d, z_d)$, such that $G(x_d, y_d, z_d).occupancy > \tau$:
   Compute the real coordinates (x, y) associated to the cell indexes (x_d, y_d).
   For $\theta_d = \theta_{dMin} \ldots \theta_{dMax}$:
   a) Compute the real value θ associated to θ_d.

   b) Compute $d = x\cos(\theta) + y\sin(\theta)$.
   c) Compute the discrete value d_d associated to d.
   d) Compute $p = x\cos(\theta + \pi/2) + y\sin(\theta + \pi/2)$.
   e) Compute the discrete value p_d associated to p.
   f) For $p'_d = p_d \ldots d_{dMax}$: increment $H(\theta_d, d_d, p'_d)$ by 1.
3) Compute $\operatorname{argmax} H_r(\alpha, d_{1\leftrightarrow 2}, d_{2\leftrightarrow 3}, d_{3\leftrightarrow 4}, d_{4\leftrightarrow 1})$.
4) Obtain the rectangle $r = (\alpha, x_c, y_c, w, h)$ using equations 10, 11, 12 and 13.

As it can be observed, the height of walls is only taken into account through histogram contributions. This is because walls correspond to 2D segment with higher histogram values than any other plane perpendicular to the floor. Thus, it is not necessary to explicitly consider the height of points in the room detection method.

Even though this method is computationally expensive, in practice, its complexity can be significantly reduced in two ways. Firstly, instead of computing H from the whole occupancy grid, it can be updated using only those cells whose occupancy state has changed. In addition, it is not necessary to apply step 3 over the entire parameter space, since only rectangles of certain dimensions are considered rooms. Thus, it is assumed a specific rank of w and h that limits the search to those values of $d_{1\leftrightarrow 2}$, $d_{2\leftrightarrow 3}$, $d_{3\leftrightarrow 4}$ and $d_{4\leftrightarrow 1}$ fulfilling that rank.

B. Door detection

The proposed 3D Hough space is also used for door detection. Doors are free passage zones that connect two different rooms, so they can be considered as empty segments of the corresponding room rectangle (i.e. segments without points). Taking this into account, once the room model is obtained, doors can be detected by analyzing each wall segment in the 3D Hough space. Therefore, for each segment of the rectangle, defined by V_i and V_j, two points $D_k = (x_k, y_k)$ and $D_l = (x_l, y_l)$ situated on the inside of that segment constitute a door segment if they verify:

$$H_{k\leftrightarrow l} = |H(\theta_{i\leftrightarrow j}, d_{i\leftrightarrow j}, p_k) - H(\theta_{i\leftrightarrow j}, d_{i\leftrightarrow j}, p_l)| = 0 \quad (14)$$

being $\theta_{i\leftrightarrow j}$ and $d_{i\leftrightarrow j}$ the parameters of the straight line defined by V_i and V_j and p_k and p_l the *signed* lengths of the segments for D_k and D_l:

$$p_k = x_k\cos(\theta_{i\leftrightarrow j} + \pi/2) + y_k\sin(\theta_{i\leftrightarrow j} + \pi/2) \quad (15)$$

$$p_l = x_l\cos(\theta_{i\leftrightarrow j} + \pi/2) + y_l\sin(\theta_{i\leftrightarrow j} + \pi/2) \quad (16)$$

Assuming $p_i \leq p_k < p_l \leq p_j$ and a minimum length l for each door segment, the door detection process can be carried out by verifying equation 14 for every pair of points between V_i and V_j, such that $p_l - p_k \geq l$. Starting from the discrete representation of the Hough space, this process can be summarized in the following steps:

1) Compute the discrete value θ_d associated to θ_{i-j}.
2) Compute the discrete value d_d associated to d_{i-j}.
3) Compute the discrete value p_{di} associated to p_i.
4) Compute the discrete value p_{dj} associated to p_j.
5) Compute the discrete value l_d associated to l (minimum length of doors).
6) $p_{dk} \leftarrow p_{di}$
7) While $p_{dk} \leq p_{dj} - l_d$:
   a) $p_{dl} \leftarrow p_{dk} + 1$
   b) While $p_{dl} < p_{dj}$ and $|H(\theta_d, d_d, p_{dk}) - H(\theta_d, d_d, p_{dl})| = 0$: $p_{dl} \leftarrow p_{dl} + 1$
   c) If $p_{dl} - p_{dk} > l_d$:
      i) Compute the real value p_k associated to p_{dk}.
      ii) Compute the real value p_l associated to $(p_{dl} - 1)$.
      iii) Compute the door limits D_k and D_l from p_k and p_l.
      iv) Insert the new door segment with extreme points D_k and D_l to the list of doors.
   d) $p_{dk} \leftarrow p_{dl}$

IV. Incremental modeling of the environment

Building topological maps requires to endow the robot not only with modeling skills, but also with the ability to actively explore the environment. Exploration plays an important role in our proposal because the robot must make sure that each room model corresponds to a real room before leaving it behind. For this reason the robot must scan the whole space surrounding it to take correct decisions about the current model.

In our system, the exploration task is driven by the 3D local grid. When a room model is detected from the set of points stored in the grid, the robot must verify it by scanning the unexplored zones inside the estimated room model. Depending on its occupancy value, the cells of the grid are labeled as *occupied*, *empty* and *unexplored*. Thus, by analyzing the grid, the robot can direct its sensor towards new places and retrieve the necessary information to get a reliable model. At this point, the benefits of using a long range sensor become clear. Few movements of the robot are needed to cover the whole space around it and, in consequence, the modeling process is less sensitive to odometric errors.

Once the local space around the robot has been completely explored, the current room model is inserted as a node in the graph representing the topological space and the robot gets out of the room to model new places. Each node in the graph stores the geometric parametrization of the room and its doors. The center and orientation of the room are used to form a reference frame (F_r) that expresses the location of the room in relation to a global reference frame (F_w). Thus, being $r = (\alpha, x_c, y_c, w, h)$ the rectangle that models a given room, the transformation matrix (T_r) that relates F_r with F_w is defined as:

$$T_r = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) & 0 & x_c \\ \sin(\alpha) & \cos(\alpha) & 0 & y_c \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (17)$$

Using this transformation matrix, any point of the model is expressed in coordinates relative to the room. This way, if the local or global reference frames are modified, points of room models remain unaffected. In addition, the robot can be aware of errors in its odometric estimation by detecting changes in the room reference frame. These changes come from the estimation of new room models, with the same dimensions than the current one, using the set of recently perceived points. Thus, being $r(i) = (\alpha(i), x_c(i), y_c(i), w, h)$ the room model at instant i and $r(i{+}1) = (\alpha(i{+}1), x_c(i{+}1), y_c(i{+}1), w, h)$ a new estimation of the room model at $i{+}1$, changes in the robot pose can be computed using the rotational and translational model deviations of equations 18 and 19.

$$\triangle\alpha = \alpha(i+1) - \alpha(i) \tag{18}$$

$$\triangle t = \begin{pmatrix} x_c(i+1) \\ y_c(i+1) \\ 0 \end{pmatrix} - \begin{pmatrix} \cos(\triangle\alpha) & -\sin(\triangle\alpha) & 0 \\ \sin(\triangle\alpha) & \cos(\triangle\alpha) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_c(i) \\ y_c(i) \\ 0 \end{pmatrix} \tag{19}$$

This is useful once the model is created for dealing with the problem of robot pose estimation. However, odometric errors are present during the whole modeling process, affecting the resulting representation in two ways. Firstly, when the robot models a new room, the location of the new room could not match its real location. Secondly, odometric errors lead to wrong measurements that cause imperfect estimations of the parameters of rooms and doors.

Though the essence of these two problems is slightly different, both can be detected and corrected using the notion of adjacent rooms. Therefore, if two adjacent rooms, $r1$ and $r2$, are communicated by a door, any point of the door is common to both rooms. Assume a door point d_{r1} viewed from the room $r1$ and the corresponding point d_{r2} of the room $r2$. The metric representation of both rooms would ideally be subject to the following restriction:

$$\|T_{r2} d_{r2} - T_{r1} d_{r1}\|^2 = 0 \tag{20}$$

When the robot creates a new room model that is adjacent to a previous one, expression 20 determines the need for correcting the new model reference frame. In addition, the deviation between the positions of the common door allows computing how this correction should be applied in order to fulfill the restriction imposed by the expression above.

After an exploration of arbitrary length, if the robot returns to a previously visited room (i.e. a *loop closing* is detected), the non-correspondence between the input and output doors can also be determined using expression 20. In such cases, new corrections must be done in order to cancel the error. However, this error is caused by wrong estimations of room and door models and, in consequence, a reference frame correction will surely not solve the problem. Our solution to these situations is to minimize a global error function defined over the whole metric representation. In our representation of the environment, the global error is defined in terms of deviations between the positions of the doors connecting adjacent rooms. Thus, the error function to minimize can be expressed as:

$$\xi = \sum_{\forall connected(d_{ri}^{(n)}, d_{rj}^{(m)})} \|T_{ri} d_{ri}^{(n)} - T_{rj} d_{rj}^{(m)}\|^2 \tag{21}$$

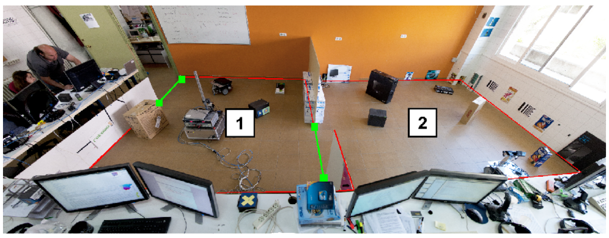
being $d_{ri}^{(n)}$ and $d_{rj}^{(m)}$ the middle points of a common door expressed in the reference frames of rooms ri and rj, respectively, and T_{ri} and T_{rj} the transformation matrices of such rooms.

The employed minimization method is based in the Stochastic Gradient Descent [12]. The basic idea [10] is to minimize the global error function by introducing small variations in the parameters of room and door models. These variations are constrained by the uncertainty of the measurement, so *high-confident* parameters remain almost unchanged during the error minimization process. As result, a reliable representation of the environment that maintains the restrictions imposed by the real world is obtained.

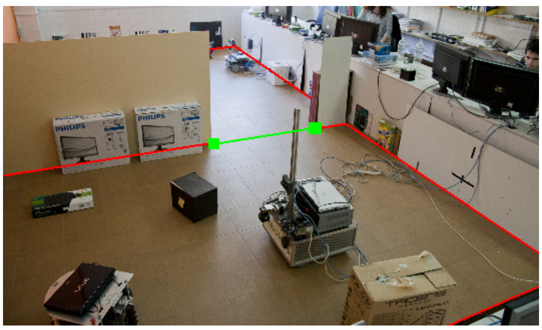
V. Experimental results

We used one of our custom differential robots to build a model of the environment. It has been equipped with a Hokuyo UTM-30LX 2D LR moved by a step motor. It scans up to 30 meters and a range of 270° at a rate of 25ms per line of scan (700 points). The LRF is positioned in the front of the robot and it performs a roll movement to cover an amplitude of 360°, (see figure 3(b)). The software has been developed using the component oriented robotics framework RoboComp [7].

The experimental results presented in this paper correspond to the modeling of an environment formed by two contiguous rooms with its corresponding doors. Rooms have been made to be similar to any regular room and have been filled with random objects to simulate a real human environment. Figure 3 shows two views of the environment used in this experiment. In these views, the two rooms, labeled as 1 and 2, and the doors have been marked in red and green, respectively.



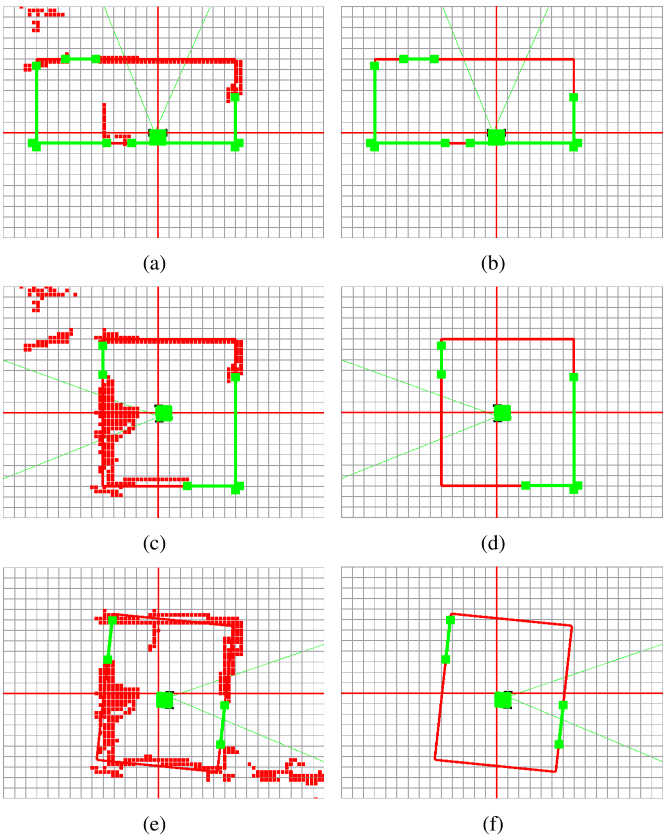
(a) Frontal view of the scene.



(b) Side view of the scene taken from the left up corner of 3(a).

Fig. 3. Two views of the real scene of the experiment.

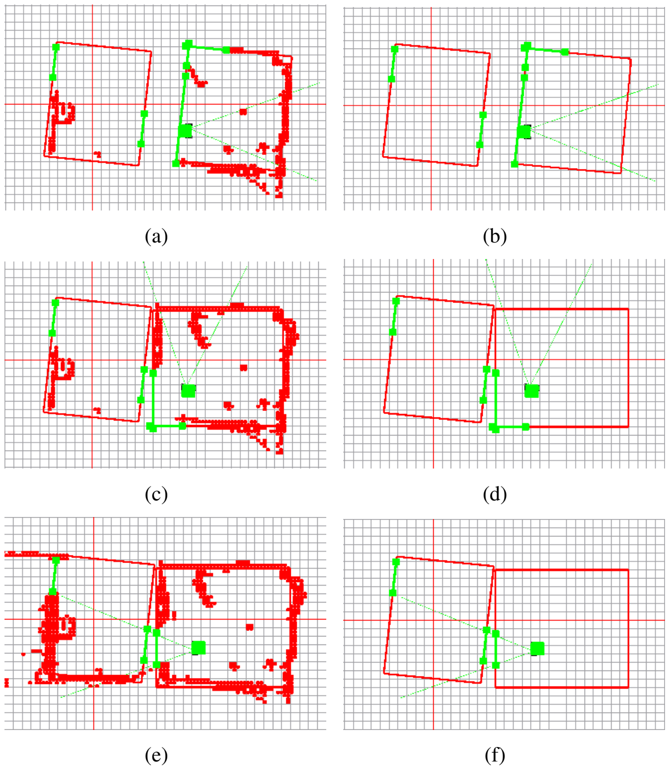
We placed our mobile robot in the center of room 1 in order to start our testing. In the first part of the experiment, the robot performs a full 3D scan of the first room. Results can be seen in figure 4. From its initial location, the robot makes a first scan and obtains an initial model of the room (figures 4(a) and 4(b)). Figure 4(a) shows the regions and the model while figure 4(b) shows only the model. Notice that regions containing points are considered walls (in red) and those without points are considered doors (in green). Using this initial model, the robot tries now to scan the unknown areas. After moving 90° counter clockwise, new points are obtained from a second scan. Figures 4(c) and 4(d) show on regions and model results, respectively. As it can be observed, the accuracy of the model increases with this second scan. The detected rectangle fits the room size but there are still some unknown parts considered as doors. The robot tries to verify whether those regions are real doors or not by moving 180° and performing another scan. After this new scan, the room is perfectly matched by the model, as figures 4(e) and 4(f) show. The room scanning is finished when all regions in the model have been scanned, and further scanning (even using higher resolutions) results in no modifications. Then, the obtained room model is fixed and stored in the topological graph. Notice that, although the model size and doors are properly obtained, the room has been a little mispositioned when compared to reality. This is due to the accumulation of odometric errors during the robot movements. Nevertheless, the relative position of the robot inside the room remains correct and therefore the detection of a new room is not affected by this misplacement.



(a) (b) (c) (d) (e) (f)

Fig. 4. Modeling process of room 1.

After the model of the first room has been fixed, the robot uses one of the obtained doors to get to the next room. In the experiment, the robot goes through the door on the right to room 2 and performs a full 3D scan. Figure 5 shows results of the second room scans. Specifically figures 5(a) and 5(b) correspond to the first one. As it can be observed, the model is still not matching the real room because of the existence of big spaces with missing information. Therefore the robot turns and get another scan of one of these places. Figures 5(c) and 5(d) show the model and regions after the second scan in this room. Now the model fits the room but, again there are still unscanned parts. The robot turns 90° again and performs the scan whose regions and resulting model are shown in figures 5(e) and 5(f). After doing all these scans in the second room the final model is obtained. Further scans do not change the model, so it is fixed.



(a) (b) (c) (d) (e) (f)

Fig. 5. Modeling process of room 2.

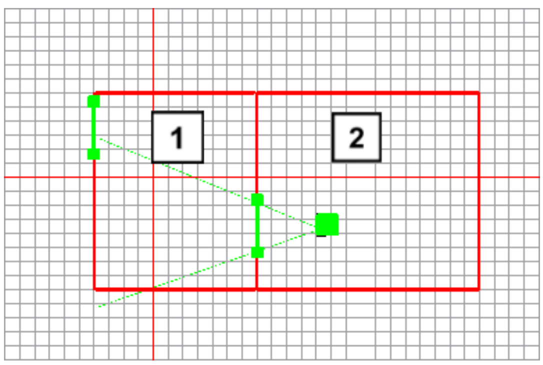


Fig. 6. Final representation obtained after the modeling of the two rooms.

Once the second model has been fixed, the deviation between the two room models (figure 5(f)), caused by odometric errors, is corrected according to the common door restriction (equation 20). Figure 6 shows the result of this correction. Each square of the figure represents an area of $0,27 \times 0,27m^2$. This size corresponds to the sampling step of the Hough space for the parameters d and p. This means that the accuracy of each room model is limited by this sampling step. The real sizes of the two rooms are $3,19 \times 3,78m^2$ (room 1) and $4,20 \times 3,78m^2$ (room 2). The sizes obtained by the modeling process are $2,97 \times 3,78m^2$ (room 1) and $4,05 \times 3,78m^2$ (room 2). As it can be observed, the difference between the representation and the real world is in the range of the permissible error. More accurate models can be obtained by reducing the sampling step of the discrete Hough space.

VI. Conclusions and Future Work

In this paper, we have presented an incremental modeling method for building hybrid representations of indoor environments. The proposed representation consists in a topological graph that describes the rooms of the environment and their connections. Each node of the graph represents a room and contains the geometric parametrization of the corresponding room and its doors. This geometric parametrization is the only metric information included in the representation. Using this information, the robot can recover a metric map of a particular zone of the environment when it is needed. Thus, no dense metric map is mantained in parallel to the topological graph. Rooms and doors are modeled using a variation of the Hough Transform that detects segments and rectangle patterns. We have proposed methods for dealing with odometric errors in the creation of new models as well as in loop closings. In addition, a method for robot pose estimation using room models has been presented. Real experiments have been carried out using a mobile robot equipped with a motorized 2D Laser. Results show the accuracy of the modeling process in real environments.

We are working on several improvements of our proposal. In particular, work in order to relax the rectangle assumption is currently in progress. We are also extending the modeling ability of our robots for representing other structures of man-made environments like corridors. Bigger and more complex possible surroundings are also being taken into account.

Acknowledgment

This work has partly been supported by grants PRI09A037 and PDT09A044, from the Ministry of Economy, Trade and Innovation of the Extremaduran Government, and by grants TSI-020301-2009-27 and IPT-430000-2010-2, from the Spanish Government and the FEDER funds.

References

[1] The primesensor technology. *http://www.primesense.com/*.

[2] R.O. Duda and P.E. Hart. Use of the hough transformation to detect lines and curves in pictures. *Commun. ACM*, 15:11–15, January 1972.

[3] Kwangro Joo, Tae-Kyeong Lee, Sanghoon Baek, and Se-Young Oh. Generating topological map from occupancy grid-map using virtual door detection. In *IEEE Congress on Evolutionary Computation*, pages 1–6, 2010.

[4] C.R. Jung and R. Schramm. Rectangle detection based on a windowed hough transform. In *Proceedins of the XVII Brasilian Symposium on Computer Graphics and Image Processing*, pages 113–120, 2004.

[5] D. Lagunovsky and S. Ablameyko. Straight-line-based primitive extraction in grey-scale object recognition. *Pattern Recognition Letters*, 20(10):1005–1014, 1999.

[6] C. Lin and R. Nevatia. Building detection and description from a single intensity image. *Computer Vision and Image Understanding*, 72(2):101–121, 1998.

[7] L. Manso, P. Bachiller, P. Bustos, P. Núñez, R. Cintas, and L. Calderita. Robocomp: a tool-based robotics framework. In *Proceedings of the Second international conference on Simulation, modeling, and programming for autonomous robots*, SIMPAR'10, pages 251–262, 2010.

[8] J. Matas, C. Galambos, and J. Kittler. Robust detection of lines using the progressive probabilistic hough transform. *Computer Vision and Image Understanding*, 78(1):119–137, 2000.

[9] E. Montijano and C. Sagues. Topological maps based on graphs of planar regions. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1661–1666, October 2009.

[10] E. Olson. *Robust and Efficient Robotic Mapping*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, June 2008.

[11] P.L. Palmer, J. Kittler, and M. Petrou. Using focus of attention with the hough transform for accurate line parameter estimation. *Pattern Recognition*, 27(9):1127–1134, 1994.

[12] H. Robbins and S. Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407, 1951.

[13] A. Rosenfeld. Picture processing by computer. *ACM Comput. Surv.*, 1:147–176, September 1969.

[14] S. Simhon and G. Dudek. A global topological map formed by local metric maps. In *In IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1708–1714, 1998.

[15] W.-B. Tao, J.-W. Tian, and J. Liu. A new approach to extract rectangular building from aerial urban images. In *Signal Processing, 2002 6th International Conference on*, volume 1, pages 143 – 146, aug. 2002.

[16] S. Thrun. Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, 99(1):21–71, 1998.

[17] N. Tomatis, I. Nourbakhsh, and R. Siegwart. Hybrid simultaneous localization and map building: a natural integration of topological and metric. *Robotics and Autonomous Systems*, 44(1):3–14, 2003.

[18] D. Van Zwynsvoorde, T. Simeon, and R. Alami. Incremental topological modeling using local vorono-like graphs. In *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and System (IROS 2000)*, volume 2, pages 897–902, 2000.

[19] F. Yan, Y. Zhuang, and W. Wang. Large-scale topological environmental model based particle filters for mobile robot indoor localization. *Robotics and Biomimetics, IEEE International Conference on*, 0:858–863, 2006.

[20] Y. Zhu, B. Carragher, F. Mouche, and C.S. Potter. Automatic particle detection through efficient hough transforms. *IEEE Trans. Med. Imaging*, 22(9), 2003.