

Probabilistic Mapping by Fusion of Range-Finders Sensors and Odometry

Anderson Souza, Adelardo Medeiros and Luiz Gonçalves
Federal University of Rio Grande do Norte
Natal, RN, Brazil

André Santana
Federal University of Piauí
Teresina, PI, Brazil

1. Introduction

One of the main challenges faced by robotics scientists is to provide autonomy to robots. That is, according to Medeiros (Medeiros, 1998) a robot to be considered autonomous must present a series of abilities as reaction to environment changes, intelligent behavior, integration of data provided by sensors (sensor fusion), ability for solving multiple tasks, robustness, operation without failings, programmability, modularity, flexibility, expandability, adaptability and global reasoning. Yet in the context of autonomy, the navigation problem appears. As described in Fig. 1, sense, plan and act capabilities have to be previously given to a robot in order to start thinking on autonomous navigation. These capabilities can be divided into sub-problems abstracted hierarchically in five levels of autonomy: *Environment Mapping*, *Localization*, *Path Planning*, *Trajectory Generation*, and *Trajectory Execution* (Alsina et. al., 2002).

At the level of *Environment Mapping* the robot system has to generate a computational model containing the main structural characteristics of the environment. In other words, it is necessary to equip the robot with sensing devices that allow the robot to perceive its surrounds acquiring useful data to producing information for construction of the environment *map*. Further, in order to get a trustworthy mapping, the system needs to know the position and orientation of the robot with relation to some fixed world referential. This process that includes sensory data capture, position and orientation inferring, and subsequently processing with objective of construction of a computational structure representing the robot underlying space is simply known as *Robotic Mapping*.

In this work, we propose a mapping method based on probabilistic robotics, with the map being represented through a modified occupancy grid Elfes (1987). The main idea is to let the mobile robot construct its surroundings geometry in a systematic and incremental way in order to get the final, complete map of the environment. As a consequence, the robot can move in the environment in a safe mode based on a trustworthiness value, which is calculated by its perceptual system using sensory data. The map is represented in a form that is coherent

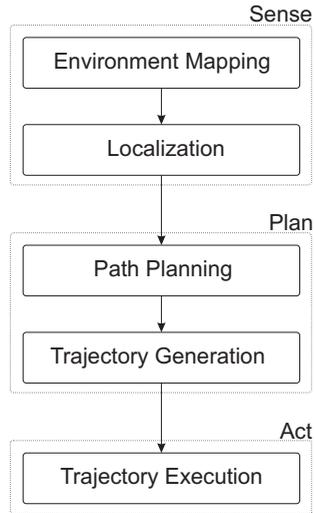


Fig. 1. Hierarchical levels for autonomous navigation.

with sensory data, noisy or not, coming from sensors. Characteristic noise incorporated to data is treated by probabilistic modeling in such a way that its effects can be visible in the final result of the mapping process. Experimental tests show the viability of this methodology and its direct applicability to autonomous robot tasks execution, being this the main contribution of this work.

In the following, the formal concepts related to robotics mapping through sensor fusion are presented. A brief discussion about the main challenges in environment mapping and their proposed solutions is presented as well the several manners of representing the mapped environments. Further, the mapping algorithm proposed in this work, based on a probabilistic modeling on occupancy grid. The proposed modeling of sensory information with fusing of sonar data that are used in this work and odometry data provided by the odometry system of the robot is described. Given that odometry is susceptible to different sources of noise (systematic and/or not), further efforts in modeling these kinds of noises in order to represent them in the constructed map are described. In this way, the mapping algorithm results in a representation that is consistent with sensory data acquired by the robot. Results of the proposed algorithm considering the robot in an indoor environment are presented and, finally, conclusions showing the main contributions and applications plus future directions are given.

2. Robotic Mapping

In order to formalize the robotics mapping problem, some basic hypotheses are established. The first is that the robot precisely knows its position and orientation inside the environment in relation to some fixed reference frame, that is, a global coordinate system. This process of inferring position and orientation of the robot in an environment is known as the localization problem. The second hypothesis is that the robot has a perceptual system, that is, sensors that

makes possible acquisition of data, proper and of the environment, such as cameras, sonars and motor encoders, between others.

With these assumptions, robotics mapping can be defined as the problem of construction of a spatial model of an environment through a robotic system based on accurate knowledge of position and orientation of the robot in the environment and on data given by the robot perceptual system.

With respect to the model used for representing the map, Thrun (Thrun, 2002) proposes a classification following two main approaches, the *topological* and the *metric* maps. Topological maps are those computationally (or mathematically) represented by way of a graph, which is a well known entity in Math. In this representation, in general, the nodes correspond to spaces or places that are well defined (or known) and the links represent connectivity relations between these places. Metric maps (or metric representations) reproduce with certain degree of fidelity the environment geometry. Objects as walls, obstacles and doorway passages are easily identified in this approach because the map has a topographic relation very close to the real world. This proposed classification is the most used up to date, besides a subtle variation that adds a class of maps based on *features* appears in some works (Choset & Fox, 2004; Rocha, 2006). This category is treated sometimes as a sub-category of the metric representation due to the storage of certain notable objects or features as for example edges, corners, borders, circles and other geometric shapes that can be detected by any feature detector.

Fig. 2 illustrates the above mentioned ways of representing a mapped environment. Each one of these forms of representation have its own advantages and disadvantages. It is easier to construct and to maintain a map based on the metric approach. It allows to recognize places with simple processing and facilitates the computation of short paths in the map. However, it requires high computational efforts to be kept and needs to know the precise position and orientation of the robot at all time, what can be a problem. On its turn, the topological representation needs few computational efforts to be kept and can rely on approximated position and orientation, besides being a convenient way for solving several classes of high-level problems. However, it is computationally expensive to construct and maintain this representation and it makes it difficult the identification or recognition of places.

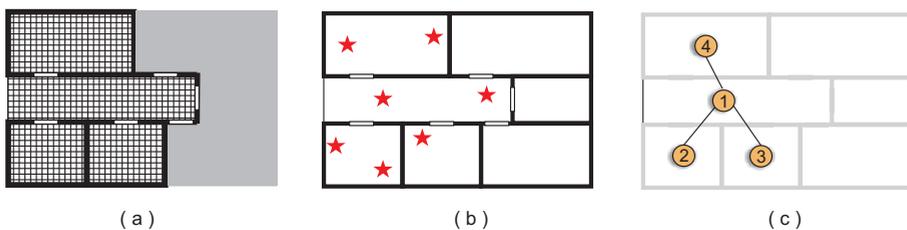


Fig. 2. (a) Metric map; (b) Feature-based map; (c) Topologic map.

Several challenges that can be found in the robotics mapping problem are enumerated by Thrun as (Thrun, 2002):

1. *Modeling sensor errors*

There are several sources of errors causing different types or natures of noise in the

sensory data. Error can be easily modeled for noises that are statistically independent in different measures. However, there is a random dependence that occurs because errors inherent to robot motion accumulate over time affecting the way that sensory measures are interpreted.

2. *Environment dimension*

Besides the lack of precision in the robot system, a second challenge is the size of the environment to be mapped, that is, the map gets less precise and more expensive to build it as the environment gets bigger.

3. *Data association*

This problem is also known as data correspondence (or matching). During the mapping, it is often current that the same object or obstacle is perceived several times by the robot system in different instants. So, it is desirable that an already seen object gets recognized and treated in a different manner than a not yet mapped object. Data association aims to determine the occurrence of this case in an efficient manner.

4. *Environment dynamics*

Another challenge is related to the mapping of dynamic environments as for example places where people are constantly walking. The great majority of algorithms for mapping considers the process running in static environments.

5. *Exploration strategy*

The mapping must incorporate a good exploration strategy, which should consider a partial model of the environment. This task appears as the fifth challenge for the robotics mapping problem.

Robots can be used to construct maps of indoor (Ouellette & Hirasawa, 2008; Santana & Medeiros, 2009; Thrun et. al., 2004), outdoor (Agrawal et. al., 2007; Triebel et. al., 2006; Wolf et. al., 2005), subterranean (Silver et. al., 2004; Thrun et. al., 2003), and underwater environments (Clark et. al., 2009; Hogue & Jenkin, 2006). With respect to its use, they can be employed in execution of tasks considered simple such as obstacle avoidance, path planning and localization. Map can also be used in tasks considered of more difficulty as exploration of galleries inside coal-mines, nuclear installations, toxic garbage cleanliness, fire extinguishing, and rescue of victims in disasters, between others. It is important to note that these tasks can be extended to several classes of mobile robots, as aerial, terrestrial and aquatic (Krys & Najjaran, 2007; Santana & Medeiros, 2009; Steder et. al., 2008).

3. Probabilistic Occupancy Grid Mapping

Errors present after acquisition process may lead to a wrong interpretation of sensory data and the consequently construction of a not reliable (Thrun, 2002). So, a treatment of these errors should be done in order to eliminate or to have at least controlled these errors. Here we choose to explore the use of a probabilistic approach in order to model these errors. Note that by knowing the amount and type of errors of a robot system, one can rely on this to let it do tasks in a more efficient way.

3.1 Localization

As explained previously, localization, that is, inferring position and orientation of the robot inside its environment is an important requirement for map construction. Some researchers makes this assumption farther important stating that localization is the fundamental and main

problem that should be treated in order to give autonomy to a robot (Cox, 1991). As well, Thrun (Thrun et. al., 2000) treats localization as the key problem for the success of an autonomous robot.

Localization methods generally fall in one of the following approaches: relative, absolute or using multisensor-fusion. Relative localization (or *dead reckoning*) is based on the integration of sensory data (generally from encoders) over time. The current localization of the robot is calculated after a time interval from the previous one plus the current displacement/rotation perceived in that time slot by the sensor. Several sources may generate errors between each time step, so, note that this approach also integrates errors. The calculated localizations are in reality estimations whose precision depends on the amount of accumulated error. In fact, the robot may be lost after some time. Absolute localization gives the actual localization of the robot at a given time. This actual localization is generally calculated based on the detection of objects or landmarks in the environment, with known localization, from which the position and orientation of the robot can be calculated by triangulation or some other methodology. Note that a GPS (and/or compassing) or similar method can also be used to get absolute position and orientation of the robot in the environment. Multi-sensor fusion combines relative and absolute localization. For example, a robot relying on its encoders may, after certain period of time, do absolute localization in order to rectify its actual localization from landmarks in the environment. In general Kalman filter and/or similar approaches are used in this situation to extend the maximum possible the amount of time necessary for absolute re-localization, since this is generally time consuming so the robot does anything while actually localizing itself. We consider using relative localization in this work since no information with respect to the environment is given to the robot previously.

One of the most used ways for estimating the robot position and orientation is by using *odometry*. Odometry gives an estimate of the current robot localization by integration of motion of the robot wheels. By counting pulses generated by encoders coupled to the wheels axes (actually, rotation sensors that count the amount of turns) the robot system can calculate the linear distance and orientation of the robot at the current instant. Odometry is most used because of its low cost, relative precision in small displacements and high rate of sampled data (Borenstein et. al., 1996). However, the disadvantage of this method is the accumulation of errors that increases proportionally to the displacement. Propagated error is systematic or not. Systematic errors are due to uncertainty in the parameters that are part of the kinematics modeling of the robot (different wheel diameters, axis length different from its actual size, finite sample rate of the encoders, and others). Non systematic errors occur due to unexpected situations as unexpected obstacles or slipping of the wheels (Santana, 2007).

Particularly, with the objective of modeling the odometry of our robot, a methodology based on utilization of empirical data (Chenavier & Crowley, 1992) is used in this work. From experimental data collected in several samples it was possible to devise the function that approximates the odometry errors. This practical experiment is done in two phases. In the first one the angular and linear errors were modeled in a linear displacement (translation only) and in the second one the angular and linear errors were modeled in an angular displacement (rotation only). From these experiments, it was possible to establish a function that describes, in approximation, the behavior of systematic errors present at the odometry system. Equations 1 and 2 represents these functions (linear and angular, respectively).

$$E_{lin}(\Delta l) = 0.09\Delta l + \sigma \quad (1)$$

$$E_{ang}(\Delta\theta) = 0.095\Delta\theta + \alpha \quad (2)$$

In the above Equations, Δl is the linear displacement estimated by odometry, $\Delta\theta$ is the angular displacement estimated by odometry, σ is the mean linear error due to a rotation and α is the mean angular error due to a linear displacement. In the performed experiments, σ and α have presented, approximately, constant values not varying proportionally with the linear and angular displacements. With the same empirical data and adopting, again, the methodology described above (Chenavier & Crowley, 1992), factors were estimated that multiplied by the linear and angular displacements gives an estimate of the variance of the non systematic errors. In this case, the errors are represented by normal distributions, or Gaussians with mean equals to zero and variance ε_{lin} for the linear case and ε_{ang} for the angular case. Equations 3 and 4 describe the computation of the variance of the linear and angular errors respectively.

$$\varepsilon_{lin} = \kappa_{ll}\Delta l + \kappa_{l\theta}\Delta\theta \quad (3)$$

$$\varepsilon_{ang} = \kappa_{\theta\theta}\Delta\theta + \kappa_{\theta l}\Delta l \quad (4)$$

κ_{ll} is the linear error coefficient in a linear displacement Δl , $\kappa_{l\theta}$ is the linear error coefficient caused by a rotation $\Delta\theta$, $\kappa_{\theta\theta}$ is the angular error coefficient in a rotation $\Delta\theta$, and $\kappa_{\theta l}$ is the angular error coefficient caused by a linear displacement Δl . These coefficients are calculated by Equations 5, 6, 7, and 8, respectively.

$$\kappa_{ll} = \frac{Var(e_{lin})}{\mu(\Delta l)} \quad (5)$$

$$\kappa_{l\theta} = \frac{Var(e_{lin})}{\mu(\Delta\theta)} \quad (6)$$

$$\kappa_{\theta\theta} = \frac{Var(e_{ang})}{\mu(\Delta\theta)} \quad (7)$$

$$\kappa_{\theta l} = \frac{Var(e_{ang})}{\mu(\Delta l)} \quad (8)$$

In Equations 5, 6, 7, and 8, parameter $Var(\cdot)$ is the variance, $\mu(\cdot)$ is the mean, e_{lin} and e_{ang} are the linear and angular errors, respectively that are obtained from the comparison between the real displacement values and the estimated given by the odometry system. By grouping the two error sources (systematic and not), a model for global error is obtained as given by Equations 9 and 10).

$$\mathbb{E}_{lin} = E_{lin}(\Delta l) + \mathcal{N}(0, \varepsilon_{lin}) \quad (9)$$

$$\mathbb{E}_{ang} = E_{ang}(\Delta\theta) + \mathcal{N}(0, \varepsilon_{ang}) \quad (10)$$

In the above Equations, $\mathcal{N}(0, \varepsilon_{lin})$ is a Gaussian noise with mean equals 0 and variance ε_{lin} to the linear case, and $\mathcal{N}(0, \varepsilon_{ang})$ is a Gaussian noise with mean equals 0 and variance ε_{ang} to the angular case. The modeling of these errors makes it possible to represent them in the environment map, resulting in a more coherent with the sensory data.

3.2 Occupancy Grid Mapping

The use of occupancy grid for mapping is proposed by Elfes and Moravec (Elfes, 1987) and is better formalized in the PhD thesis of the first author (Elfes, 1989). The objective is to construct consistent maps from sensory data under the hypothesis that the robot position and orientation is known. The basic idea is to represent the environment as a grid that is a multi-dimensional matrix (in general 2D or 3D) that contains cells of the same size. Each cell corresponds to a random variable that represents its occupancy probability. Fig. 3 shows an example of a occupancy grid of part of an environment using data provided by a sonar array.

Dark cells represent objects (or obstacles) detected by the sonar array, clear cells represent free regions, and gray cells are regions not yet mapped. Spatial model based on occupancy grid can be directly used in navigation tasks, as path planning with obstacle avoidance and position estimation (Elfes, 1989). The state values are estimated by way of interpretation of data coming from depth sensors probabilistic modeled using probabilistic function. It is possible to update each cell value through Bayesian probabilistic rules every time that new readings are done in different positions in the environment.

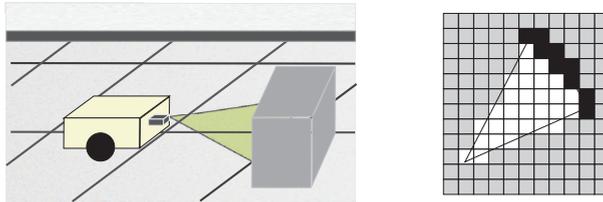


Fig. 3. Use of occupancy grid for representing sonar array data.

Most part of current researches related to environment mapping for robotics uses probabilistic techniques constructing probabilistic models for the robots, sensors and mapped environments. The reason for the popularity is of probabilistic techniques comes from the assumed existence of uncertainty present in sensory data. With probabilistic techniques, it is possible to treat this problem by explicitly modeling the several sources of noise and their influence in the measures (Thrun, 2002).

The standard algorithm formalized by Elfes (Elfes, 1989) aims to construct a map based on sensory data and knowing the robot position and orientation. In our work, we use the odometry system of the robot for calculating position and orientation. So the occupancy grid map construction is based on the fusion of data given by the sonars with data provided by the odometry system of the robot. Equation 11 presents the mathematical formulation that usually describes the occupancy grid mapping (Elfes, 1987; 1989; Thrun et. al., 2003; 2005).

$$P(\mathbf{m}|\mathbf{z}_{1:t}) \quad (11)$$

In the Equation 11, \mathbf{m} represents the acquired map and $\mathbf{z}_{1:t}$ is the set of sensory measures realized up to time instant t . It is important to clear that the algorithm assumes that position and orientation of the robot are known. Continuous space is discretized in cells that, together, approximate the environment shape. This discretization corresponds to a plan cut of the 3D environment in the case of using a 2D grid or could be a 3D discretization in the case of a 3D grid. This depends of the sensors model and characteristics. For example, sonar allows a 2D

sample of the environment, however stereo vision allows a 3D reconstruction. In this work, we use sonars.

Considering the discretization of the environment in cells, the map \mathbf{m} can be defined as a finite set of cells $m_{x,y}$ where each cell has a value that corresponds to the probability of it being occupied. The cells can have values in the interval $[0, 1]$ with 0 meaning empty and 1 meaning occupied. Being the map a set of cells, the mapping problem can be decomposed in several problems of estimation of the value of each cell in the map. Equation 12 represents an instance for the estimation of the value of a cell $m_{x,y}$, that is, the probability of cell $m_{x,y}$ being occupied when sensory measures $\mathbf{z}_{1:t}$ until the t instant.

$$P(m_{x,y}|\mathbf{z}_{1:t}) \quad (12)$$

Due to numerical instability, with probabilities close to 0 or 1 it is common to calculate the *log-odds* (or probability logarithm) of $P(m_{x,y}|\mathbf{z}_{1:t})$ instead of $P(m_{x,y}|\mathbf{z}_{1:t})$. The *log-odds* is defined by:

$$l_{x,y}^t = \log \frac{P(m_{x,y}|\mathbf{z}_{1:t})}{1 - P(m_{x,y}|\mathbf{z}_{1:t})} \quad (13)$$

The probability occupancy value can be recovered through Equation 14.

$$P(m_{x,y}|\mathbf{z}_{1:t}) = 1 - \frac{1}{e^{l_{x,y}^t}} \quad (14)$$

The value of *log-odds* can be estimated recursively at any instant t by using the Bayes rule applied to $P(m_{x,y}|\mathbf{z}_{1:t})$ (see Equation 15).

$$P(m_{x,y}|\mathbf{z}_{1:t}) = \frac{P(z_t|\mathbf{z}_{1:t-1}, m_{x,y})P(m_{x,y}|\mathbf{z}_{1:t-1})}{P(z_t|\mathbf{z}_{1:t-1})} \quad (15)$$

in Equation 15, $P(z_t|\mathbf{z}_{1:t-1}, m_{x,y})$ represents the probabilistic modelo of the depth sensor, $P(m_{x,y}|\mathbf{z}_{1:t-1})$ is the value of cell $m_{x,y}$ at instant $t - 1$ and $P(z_t|\mathbf{z}_{1:t-1})$ is the real value measured by the sensor. Assuming that the mapping is performed in random environments, the current measure of the sensor is independent of past measures, given the map \mathbf{m} at any instant. This results in Equations 16 and 17.

$$P(z_t|\mathbf{z}_{1:t-1}, \mathbf{m}) = P(z_t|\mathbf{m}) \quad (16)$$

$$P(z_t|\mathbf{z}_{1:t-1}) = P(z_t) \quad (17)$$

Given that the map is decomposed in cells, this suposition can be extended as shown in Equation 18.

$$P(z_t|\mathbf{z}_{1:t-1}, m_{x,y}) = P(z_t|m_{x,y}) \quad (18)$$

With basis on the above assumptions, Equation 15 can be simplified resulting in Equation 19.

$$P(m_{x,y}|\mathbf{z}_{1:t}) = \frac{P(z_t|m_{x,y})P(m_{x,y}|\mathbf{z}_{1:t-1})}{P(z_t)} \quad (19)$$

By applying the *total probability* rule to Equation 19, Equation 20 is obtained. The last calculates the probability of occupation for cell $m_{x,y}$ having as basis the probabilistic model of sensor $P(z_t|m_{x,y})$ and the occupancy value of the cell available previously $P(m_{x,y}|z_{1:t-1})$.

$$P(m_{x,y}|z_{1:t}) = \frac{P(z_t|m_{x,y})P(m_{x,y}|z_{1:t-1})}{\sum_{m_{x,y}} P(z_t|m_{x,y})P(m_{x,y}|z_{1:t-1})} \quad (20)$$

Computationally, the mapping using occupancy grid can be implemented by Algorithm 1 (Thrun et. al., 2005). The algorithm has as input variables a matrix with all occupancy values $\{l_{t-1,(x,y)}\}$ attributed to the occupancy grid constructed until instant $t - 1$, a robot localization vector $\mathbf{x}_t = (x, y, \theta)'$ at instant t and the values of sensor readings z_t at instant t . If a cell $m_{x,y}$ of the occupancy grid is inside the field of view of the sensors (line 2), the occupancy grid value is updated taking into account the previous value of the cell $l_{t-1,(x,y)}$, the sensor model $inverse_sensor_model(m_{x,y}, \mathbf{x}_t, z_t)$ and the constant l_0 that is attributed to all cells at beginning indicating that they are not mapped (line 3). If the cell $m_{x,y}$ is out of the field of view, its value is kept (line 5).

Algorithm 1 occupancy_grid_mapping($\{l_{t-1,(x,y)}\}, \mathbf{x}_t, z_t$)

```

1: for all cells  $m_{x,y}$  do
2:   if  $m_{x,y}$  in perceptual field of  $z_t$  then
3:      $l_{t,(x,y)} = l_{t-1,x,y} + inverse\_sensor\_model(m_{x,y}, \mathbf{x}_t, z_t) - l_0$ 
4:   else
5:      $l_{t,(x,y)} = l_{t-1,(x,y)}$ 
6:   end if
7: end for
8: return  $\{l_{t,(x,y)}\}$ 
    
```

It is important to emphasize that the occupancy values of the cells at Algorithm 1 are calculated through log-odd that is the logarithm of the probability of avoiding numerical instabilities. In order to recover the probability values Equation 14 can be used.

With basis on this algorithm, we implemented the method proposed in this work. The main difference is in the probabilistic modeling of the sensors. Our proposed model implements the $inverse_sensor_model$ used in the algorithm.

3.3 Proposed Model

In this work, we use a sonar array as sensors measuring the distance of the robot with respect to some object. Sonar arrays are often used because of its fast response time, simplicity on its output (distance is directly given) and its low cost when compared to other sensor types (Lee et. al., 2006). In general, a setup as our (array of sonar) is mounted. The used setup has an uncertainty of about 1% of the measured value and an aperture that rotates around $+15^\circ$ to -15° in relation to the main axis (see Fig. 4). The distance returned by the sonar is the one to the closest object inside its sonar beam. The maximum returned distance depends on the sonar model.

Let consider three regions inside the working area of the sonar as seen in Fig. 4. Region I represents free area. Region II is associated to the sensor measure such that the object that has

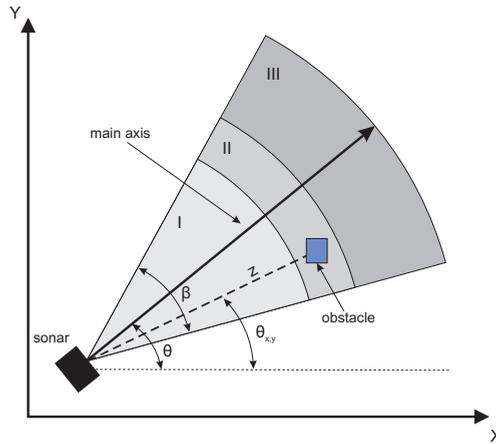


Fig. 4. Regiões significativas em feixe de sonar.

reflected the sound wave may be anywhere inside this region. Region III is the one covered, in theory, by the sonar beam. However it is not known if it is empty or occupied. Considering the above regions, the model adopted to represent the sonar is described as a Gaussian distribution as given by Equation 21.

$$P(z, \theta | d_{x,y}, \theta_{x,y}) = \frac{1}{2\pi\sigma_z\sigma_\theta} \exp \left[-\frac{1}{2} \left(\frac{(z - d_{x,y})^2}{\sigma_z^2} + \frac{(\theta - \theta_{x,y})^2}{\sigma_\theta^2} \right) \right] \tag{21}$$

In the above Equation, θ is the orientation angle of the sensor with respect to the x axis of the global reference frame (see Fig. 4), $\theta_{x,y}$ is the angle between the vector with initial point at the sonar through cell $m_{x,y}$, that may be or not with obstacle, and to the global frame axis x (see Fig. 4), σ_z^2 and σ_θ^2 are the variance that gives uncertainty in the measured distance z and in the θ angle, respectively. Fig. 5 illustrates the function that estimates the occupancy for this model.

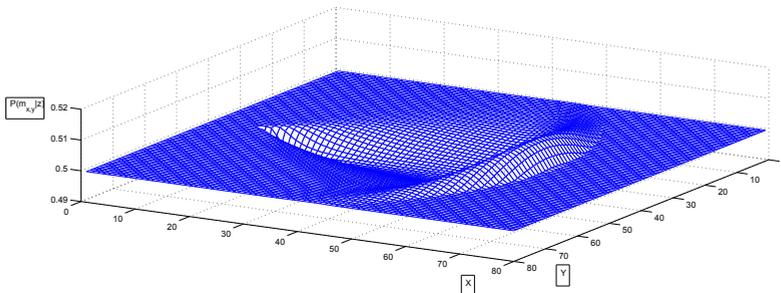


Fig. 5. Function of occupancy for a sensor modeled by a two-dimensional Gaussian distribution. Both uncertainties in the angle and in the distance being represented.

Having as basis a 2D Gaussian model in this work we also consider uncertainties that are inherent to the odometry system besides sonar uncertainties. Using odometry errors modele given in Section 3.1 of this text, described by Equations 9 and 10, it is possible to establish a relation between the variances σ_z^2 and σ_θ^2 that (model sonar errors) with odometry errors as:

$$\sigma_z = z \times \eta + E_{lin}$$

$$\sigma_\theta = \frac{\beta}{2} + E_{ang}$$

or

$$\sigma_z = z \times \eta + E_{lin}(\Delta l) + \mathcal{N}(0, \varepsilon_{lin}) \tag{22}$$

$$\sigma_\theta = \frac{\beta}{2} + E_{ang}(\Delta\theta) + \mathcal{N}(0, \varepsilon_{ang}) \tag{23}$$

In the above Equations, z the measure given by the sonar, η is an error factor typical of the sonar in use (an error of about 1%) and β the aperture angle of the sonar beam (see Fig. 4). Variances σ_z^2 and σ_θ^2 can be calculated through Equations 22 and 23, now considering the influences caused by odometry. Equation 22 calculates uncertainty to a distance z and a linear displacement Δl . $E_{lin}(\Delta l)$ is the function used to compute systematic errors of odometry (Equation 1) and $\mathcal{N}(0, \varepsilon_{lin})$ is the normal distribution used to compute non systematic errors (Equation 3). Equation 23 gives uncertainty of the orientation angle of the sonar θ and an angular displacement $\Delta\theta$ performed by the robot. $E_{ang}(\Delta\theta)$ (Equation 2) describes the systematic error of an angular displacement and $\mathcal{N}(0, \varepsilon_{ang})$ (Equation 4) is the normal distribution that estimates non systematic errors for the same displacement.

Through this proposed modification, it is possible to represent degradation of the odometry errors in the map. The probability for a correct measure calculated by Equation 21 is now weighted by the errors of the odometry system. In this way, the final map is more coherent with the quality of sensors data (sonar and odometry). Fig. 5 illustrates a degraded measure mainly by angular errors in odometry. Fig. 6 illustrates a degraded measure mainly by linear errors of odometry.

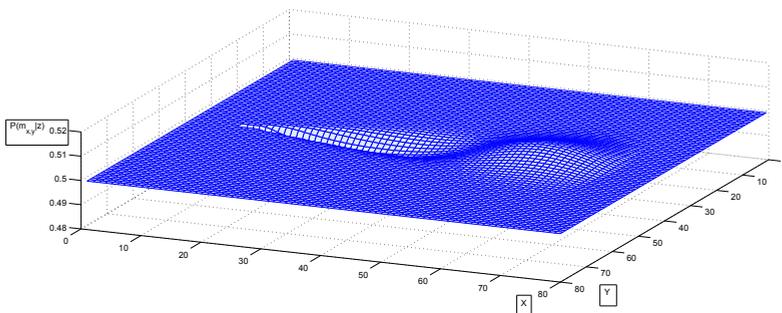


Fig. 6. Measurements degraded mainly by linear odometry errors.

3.4 Map Generation

The processing steps done in data coming from the sonars (Souza, 2008) are listed next.

- **Preprocessing:** Data coming from sonar goes through a filter that discards false readings. Distances measured below a minimum threshold and above a maximum threshold are eliminated due to its susceptibility to errors. These limits are 4 and 15 cm in this work, respectively.
- **Sensor position:** Position and orientation of the sensors with respect to the robot and also the robot with respect to the reference frame are calculated.
- **Interpretation by the probabilistic model:** Sonar data are interpreted by the proposed probabilistic model to form the *sonar view*.
- **Sonar map:** Each sonar generates its own local map from its *view* that is then added to the global map.

As exposed above, odometry errors accumulate over robot motion and degrades the quality of the map. At a certain time, the value attributed to a given cell does not have a substantial influence in the form used to define if it is occupied, empty or not mapped yet. At this instant, the mapping process is strongly corrupted and an absolute localization approach must be used to correct these errors in order for the robot to continue the mapping.

4. Experiments

The robot used in this work is a Pioneer-3AT, kindly named Galatea, which is projected for locomotion on all terrain (the AT meaning) (see Fig. 7).



Fig. 7. Galatea robotic platform.

The Pioneer family is fabricated by ActiveMedia Robotics designed to support several types of perception devices. The robot comes with an API (Application Program Interface) called ARIA (ActiveMedia Robotics Interface) that has libraries for C++, Java and Python languages (in this work we use C++ language.). The ARIA library makes possible to develop high-level programs to communicate with and control the several robot devices (sensors and actuators) allowing reading of data about the robot in execution time.

The software package comes with a robot simulator (MobileSim) that allows to test some functions without using the real robot. It is possible to construct environments of different

shapes to serve as basis to experiments and tests. Galatea has an embedded computer, a PC104+ with a pentium III processor of 800MHz, 256Mb of RAM memory, a 20Gb hard disk, communication interface RS232, Ethernet connection and wireless network board 10/100. Galatea has 2 sonar arrays with 8 sonars in which one and encoders coupled to the 2 motor axes that comprises its odometry system. To control all of these, we use the RedHat 7.3 Linux operating system.

4.1 Initial Simulated Experiments

Preliminary tests of our proposed algorithm are done with the MobileSim. We have simulated environment using a CAD model that comes with the simulator describing one of the buildings of Columbia University at USA. Fig. 8 shows the geometry of this test environment.

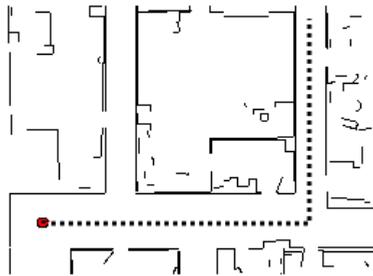


Fig. 8. Simulated environment to be mapped.

The simulated robot has mapped part of this environment following the dotted path at Fig. 8. The robot perform the mapping process until the odometry errors degrade the quality of the final map. At this point, values of each cell do not define anymore whether a cell is occupied, empty or mapped yet. That is, the robot has no subsides to construct a trustable map due to odometry errors. Part (a) of Fig. 9 illustrates this situation. Galatea is represented by the red point, white regions are empty, dark regions are occupied cells and gray regions are not mapped yet cells. As this situation occur, we simulated an absolut localization for Galatea correcting its odometry and consequently indicating that it can continue the mapping without considering past accumulated errors. Fig. 9 (b) shows the moment at which the robot localization is rectified and Fig. 9 (c) illustrates the mapping continuation after this moment.

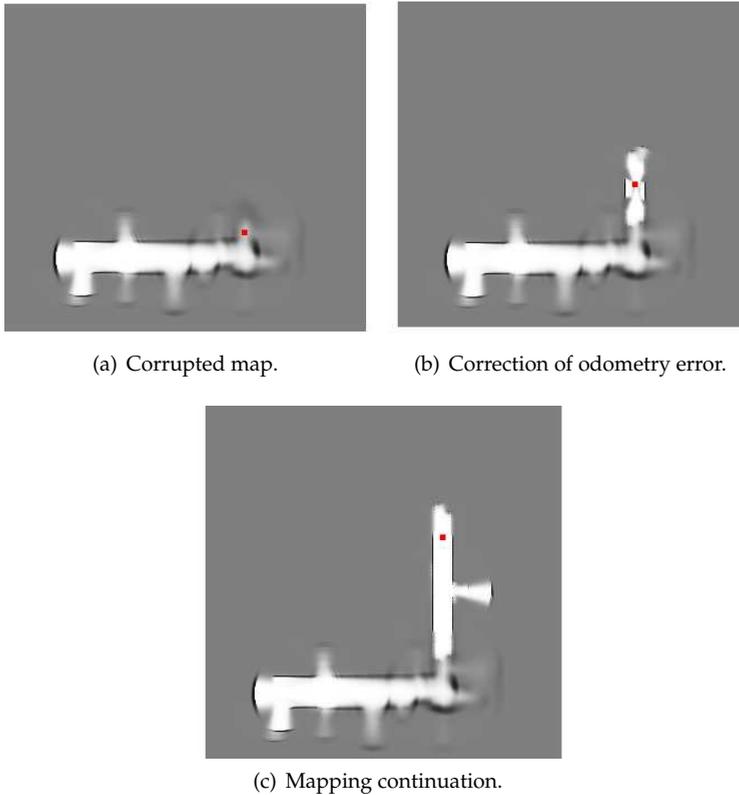


Fig. 9. Sequences of steps in the mapping of a simulated environment.

4.2 Experiments with Galatea robot

We performed several experiments at Computing Engineering Department (DCA) building at UFRN, Brazil. The building geometry is formed by corridors and rectangular rooms. Some initial experiments are shown in previous work (Souza et. al., 2008) using a simplified model for the depth sensors in order to verify the system behavior as a whole. From this initial simplified setup we could evaluate influence of problems that typical of sonars in the map quality, not possible in simulation. The main problem detected is the occurrence of multiple reflections. Fig. 10 shows a map constructed for the corridors of DCA building. Dotted line indicates the real localization of the walls. It is easy to note that several measures indicate obstacles or empty areas behind the walls plan, e.g, in areas that could not be mapped. These false measures are typically caused by multiple reflections inside the environment.

Robots with a more complex shape as the one used in this work are more susceptible to false measures. This happens because the sensors, in general, have irregular distribution along the robot body facilitating occurrence of false measures. On the opposite, robots with circular shape have a regular distribution, being easier to eliminate these false measures only by tuning the sensors characteristics (Ivanjko & Petrovic, 2005). In order to solve this problem, we have implemented a method for filtering false measures. This strategy represents the sonar

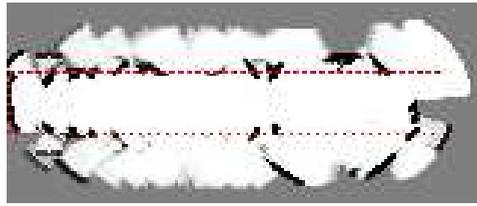


Fig. 10. Corrupted mapping due to wrong measures of the sonars.

measures by circles in such a way that if a given measure invades the circular region defined by another measure, the invaded region measure is eliminated. This technique is called Bubble Circle (BC) Threshold (Lee & Chung, 2006). Results for our work were not convincing thus other alternatives were studied.

After several experiments and observations, we could verify that in environments with rectangular shapes, as this one formed by strait corridors and not as big rooms, more consistent maps are constructed by using the side sonars. These comprise angles of 90° with the walls when the robot is parallel to the walls. The same for the rear and front sonars with smallest angles with respect to the robot main axis. So we discard the other sensors given that they produced false measures due to its disposition with respect to the walls. In fact we believe that these several other sensors with oblique angles were projected to be used when the robot is operating in outdoor environments. In fact, we could verify latter that the same consideration has been reported in the work of Ivanjko (Ivanjko & Petrovic, 2005), in this case working with the Pioneer-2DX robot.

Now, after solving the above mentioned problems, we have done another set of experiments using Algorithm 1, which is a modification of the one proposed by Thrun (Thrun et. al., 2005). The main differential of the algorithm proposed here is the inclusion of the probabilistic model of the sensor that represents the uncertainties inherent to perception in the occupancy grid map. Fig. 11 shows the mapping of the same corridor yet in the beginning of the process, however with degradation caused by the odometry error. The red dotted boundary indicates actual walls position. The red point at the white region indicates the localization of Galatea at the map and the point at the right extremity of the Fig. is the last point where the mapping should stop. Fig. 12 shows the evolution of the mapping. Observe the decreasing of quality in the mapping as the robot moves that, in its turn, increases the odometry errors. The map shown in Fig. 13 presents an enhancement in its quality. At this point, the absolute localization was done because the map was degraded. Odometry error goes to zero here, rectifying the robot position and orientation.

The mapping process goes up to the point shown in Fig. 14 where it is necessary another correction by using absolute localization, then going until the final point as shown in Fig. 15. By considering probabilistic modeling of odometry errors and sensor readings to try to diminish error effects at the mapping process, we could have a substantial enhancement in the mapping quality. However, we remark that at some time of the process the map gets even-

tually corrupted by the effects of non systematic errors. Fig. 16 shows the situation using modeling to attenuating the effect of these errors. In this case, the effect is very small because the errors become very little for the travelled distance.

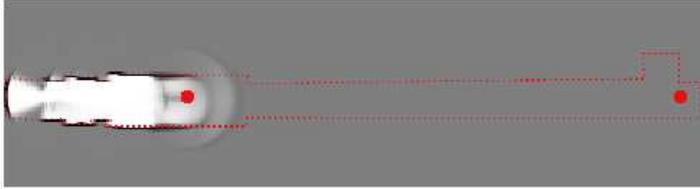


Fig. 11. Use of the proposed model considering representation of odometry errors in the mapping.

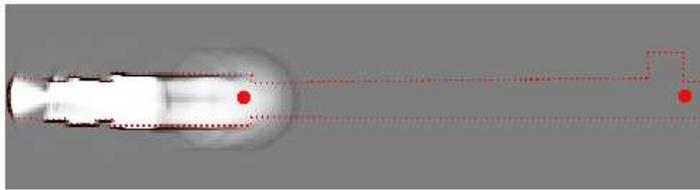


Fig. 12. Representation of odometry error in the map construction.

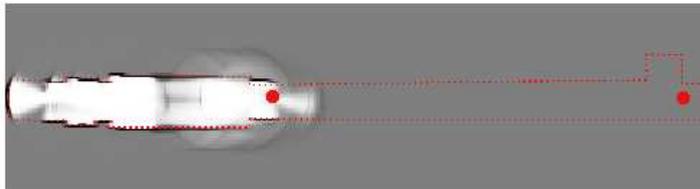


Fig. 13. Rectification of the robot localization (absolute localization).

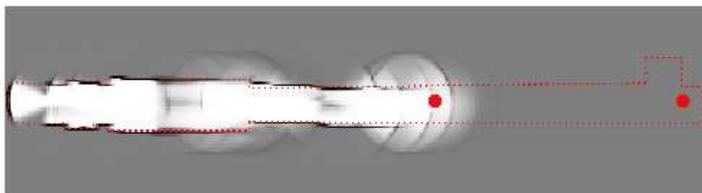


Fig. 14. Nova correção da localização.

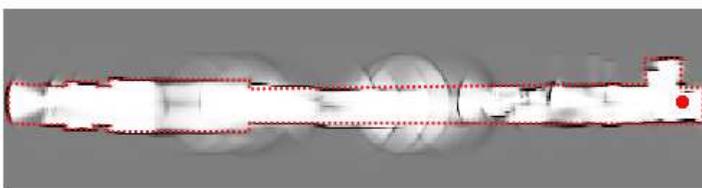


Fig. 15. Mapa final adquirido com o modelo proposto.

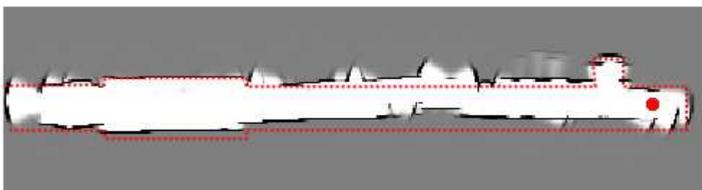


Fig. 16. Mapa construído com o modelo proposto e como correção dos erros sistemáticos de odometria.

5. Conclusion and Future Directions

In this work we propose a method for mapping with spatial representation of the environment based on occupancy grid. Our method incorporates a probabilistic model for sonar that consider the uncertainties inherent to this type of sensor as well as for the accumulative errors caused by odometry system of the robot. With this modeling, the quality of the map gets influenced by the uncertainty given by the odometry system indicating the actual trustworthiness of the sensory data collected by the robot system. Once a map gets constructed, the robot can use it to perform other high-level tasks as navigation, path planning, and decision taking, between others.

With basis on the results given by the performed experiments, we conclude that the algorithm proposed in this work gives a more realistic and actual manner for representing a mapped environment using the occupancy grid technique. This is because that now we know that the data provided by the sensors have errors and we know how much this error can grow, that is, we have an upper limit for the error, controlling its growing. Even with the difficulties given by the sonar limitations, our system presents satisfactory results. Other types of depth sensors can be added to this model or use a similar approach, as for example *lasers*, thus increasing map consistency.

As next work, we intend to study techniques and exploration heuristics in order for a robot to perform the mapping process in autonomous way. Besides, forms to enhance robot localization with incorporation of other sensors will also be studied that together can improve map quality. Future trials with emphasis in Localization and Simultaneous Mapping (SLAM) will also be done, having as basis the studies done in this work. Fusion of information with the ones provided by a visual system (stereo vision) will be further done. With this, we intend to explore the construction of 3D maps allowing the use of robots in other higher level tasks, as for example, analysis of building structure.

6. References

- Agrawal, M.; Konolige, K. & Bolles, R.C. (2007). Localization and Mapping for Autonomous Navigation in Outdoor Terrains : A Stereo Vision Approach. In: *IEEE Workshop on Applications of Computer Vision, WACV '07*.
- Alsina, P. J.; Gonçalves, L. M. G.; Medeiros, A. A. D.; Pedrosa, D. P. F. & Vieira, F. C. V. (2002). Navegação e controle de robôs móveis. In: *Mini Curso - XIV Congresso Brasileiro de Automática, Brazil*.
- Borenstein, J.; Everett, R. H. & Feng, L. (1996). *Where am I? Sensors and Methods for Mobile Robot Positioning*, University of Michigan, EUA.
- Chenavier, F. & Crowley, J. L. (1992). Position estimation for a mobile robot using vision and odometry, In: *Proceeding of the 1992 IEEE International Conference on Robotics and Automation, Nice, France*.
- Choset, H. & Fox, D. (2004). The World of Mapping. In: *Proceedings of WTEC Workshop on Review of United States Research in Robotics*, National Science Foundation (NSF), Arlington, Virginia, USA.
- Clark, C. M.; Olstad, C. S.; Buhagiar, K. & Gambin, T. (2009). Archaeology via Underwater Robots: Mapping and Localization within Maltese Cistern Systems. In: *10th International Conf. on Control, Automation, Robotics and Vision*, pp.662 - 667, Hanoi, Vietnam.
- Cox, I. J. (1991). Blanche - An Experiment in Guidance and Navigation of an Autonomous Robot Vehicle, In: *IEEE Transactions on Robotics and Automation*, Vol. 7, No. 2.
- Elfes, A. (1987). Sonar-based real-world mapping and navigation, In: *IEEE Journal of Robotics and Automation*, Vol. 3, No. 3, pp. 249-265.
- Elfes, A. (1989). Occupancy Grid: A Probabilistic Framework for Robot Perception and Navigation, *PhD Thesis*, Carnegie Mellon University, Pittsburg, Pennsylvania, USA.
- Hogue, A. & Jenkin, M. (2006). Development of an Underwater Vision Sensor for 3D Reef Mapping, In: *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2006)*, pp. 5351-5356.

- Ivanjko, E. & Petrovic, I. (2005). Experimental Evaluation of Occupancy Grids Maps Improvement by Sonar Data Correction, In: *Proceedings of 13th Mediterranean Conference on Control and Automation*, Limassol, Cyprus.
- Krys, D. & Najjaran, H. (2007). Development of Visual Simultaneous Localization and Mapping (VSLAM) for a Pipe Inspection Robot. In: *Proceedings of the 2007 International Symposium on Computational Intelligence in Robotics and Automation, CIRA 2007*. pp. 344-349.
- Lee, K. & Chung, W. K. (2006). Filtering Out Specular Reflections of Sonar Sensor Readings, In: *The 3rd International Conference on Ubiquitous Robots and Ambient Intelligent (URAI)*.
- Lee, Y.-C.; Nah, S.-I.; Ahn, H.-S. & Yu, W. (2006). Sonar Map Construction for a Mobile Robot Using a Tethered-robot Guiding System, In: *The 6rd International Conference on Ubiquitous Robots and Ambient Intelligent (URAI)*.
- Medeiros, A. A. D. (1998). A Survey of Control Architectures for Autonomous Mobile Robots. In: *JBCS - Journal of the Brazilian Computer Society, Special Issue on Robotics*, ISSN 0104-650004/98, vol. 4, n. 3, Brazil.
- Ouellette, R. & Hirasawa, K. (2008). Mayfly: A small mapping robot for Japanese office environments. In: *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pp. 880-885.
- Rocha, R. P. P. (2006). Building Volumetric Maps with Cooperative Mobile Robots and Useful Information Sharing: A Distributed Control Approach based on Entropy. *PhD Thesis*, FEUP - Faculdade de Engenharia da Universidade do Porto, Portugal.
- Santana, A. M. (2007). Localização e Planejamento de Caminhos para um Robô Humanóide e um Robô Escravo com Rodas. *Master Thesis*, UFRN, Natal, RN. 2007.
- Santana, A. M. & Medeiros, A. A. D. (2009). Simultaneous Localization and Mapping (SLAM) of a Mobile Robot Based on Fusion of Odometry and Visual Data Using Extended Kalman Filter. In: *In Robotics, Automation and Control*, Editor: Vedran Kordic, pp. 1-10. ISBN 978-953-7619-39-8. In-Tech, Austria.
- Silver, D.; Ferguson, D.; Morris, A. C. & Thayer, S. (2004). Features Extraction for Topological Mine Maps, In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2004)*, Vol. 1, pp. 773-779.
- Souza, A. A. S. (2008). Mapeamento com Grade de Ocupação Baseado em Modelagem Probabilística, *Master Thesis*, UFRN, Natal, Brazil.
- Souza, A. A. S.; Santana, A. M.; Britto, R. S.; Gonçalves, L. M. G. & Medeiros, A. A. D. (2008). Representation of Odometry Errors on Occupancy Grids. In: *Proceeding of International Conference on Informatics in Control, Automation and Robotics (ICINCO2008)*, Funchal, Portugal.
- Steder, B.; Grisetti, G.; Stachniss, C. & Burgard, W. (2008). Visual SLAM for Flying Vehicles, In: *IEEE Transactions on Robotics*, Vol. 24, Issue 5, pp. 1088-1093.
- Triebel, R.; Pfaff, P. & Burgard, W. (2006). Multi-Level Surface Maps for Outdoor Terrain Mapping and Loop Closing. In: *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2276 - 2282.
- Thrun, S. (2002). Robotic mapping: A survey. In: *Exploring Artificial Intelligence in the New Millennium*, Ed. Morgan Kaufmann, 2002.
- Thrun, S.; Fox, D.; Burgard, W. & Dellaert, F. (2000). Robust Monte Carlo Localization for Mobile Robots, In: *Artificial Intelligence*, Vol. 128, No. 1-2, pp. 99-141.
- Thrun, S.; Hähnel, D.; Ferguson, D.; Montermelo, M.; Riebel, R.; Burgard, W.; Baker, C.; Omohundro, Z.; Thayer, S. & Whittaker, W. (2003). A system for volumetric mapping

- of underground mines. In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2003)*, Vol. 3, pp. 4270-4275.
- Thrun, S.; Martin, C.; Liu, Y.; Hähnel, D.; Emery-Montemerlo, R.; Chakrabarti, D. & Burgard, W. (2004). A Real-Time Expectation Maximization Algorithm for Acquiring Multi-Planar Maps of Indoor Environments with Mobile Robots, In: *IEEE Transactions on Robotics and Automation*, Vol. 20, No. 3, pp. 433-443.
- Thrun, S.; Fox, D. & Burgard, W. (2005). *Probabilistic Robotics*, MIT Press, Cambridge.
- Wolf, D.; Howard, A. & Sukhatme, G. S. (2005). Towards geometric 3D mapping of outdoor environments using mobile robots. In: *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2005)*, pp. 1507-1512.