# Computation of the Optimal Relative Pose between Overlapping Grid Maps through Discrepancy Minimization 

Diego Rodriguez-Losada, Paloma de la Puente, Alberto Valero, Pablo San Segundo, Miguel Hernando<br>Centro de Automatica y Robotica (UPM-CSIC)<br>Madrid, Spain. Tel: 913367729 (e-mail: diego.rlosada@upm.es)


#### Abstract

Grid maps are a common environment representation in mobile robotics. Many Simultaneous Localization and Mapping (SLAM) solutions divide the global map into submaps, forming some kind of graph or tree to represent the structure of the environment, while the metric details are captured in the submaps. This work presents a novel algorithm that is able to compute a physically feasible relative pose between two overlapping grid maps. Our algorithm can be used for correspondence search (data association), but also for integrating negative information in a unified way. This paper proposes a discrepancy measure between two overlapping grid maps and its application in a quasi Newton optimization algorithm, with the hypothesis that minimizing such discrepancy could provide useful information for SLAM. Experimental evidence is provided showing the high potential of the algorithm.


Keywords: Mobile robots, Robot navigation, Optimization, Robotic mapping.

## 1. INTRODUCTION

Robotic mapping is a key component of autonomous mobile robots if true autonomy is desired. Many researchers have focused on the Simultaneous Localization and Mapping (SLAM) problem over the last couple of decades.

The probabilistic approach has dominated the solution to the SLAM problem (Thrun, 2002). The Extended Kalman Filter has probably been the most extended approach to the SLAM problem but MonteCarlo particle filter based algorithms Expectancy Maximization, hybrid approaches and even topological SLAM are also other successful solutions to the problem.

From the metric environment representation point of view two different paradigms can be found: occupancy grid mapping (Elfes 1987, Moravec and Elfes 1985) and feature based approaches. An occupancy grid divide the environment in cells, and computes the likelihood of occupancy of each individual cell, while feature based approaches represent the map as a set of individual entities with a geometric parameterization (points, segments).

Occupancy grids have some interesting advantages: they are very easily computed, no prior structure is assumed so all obstacles can be easily represented being very useful for path planning and control tasks, and are widely used in the robotic community. On the other hand, occupancy grid maps do not maintain any knowledge about the full SLAM posterior, so they are not a complete solution by themselves, being referred as a "mapping with known poses" solution. Many full SLAM solutions overcome this limitation with the use some other kind of representing the full SLAM posterior over
the robot path given the measurements with other techniques as particle filters (Hahnel et al 2003) while using grid maps as the environment representation associated to each particle.

As the environment size increases, the computational burden of monolithic approaches can become a problem. The most common approach to large scale SLAM is the divide and conquer strategy in which the map is divided into smaller submaps, so the geometric information is efficiently handled locally inside the submaps, while the environment overall structure is captured in a graph or tree like representation. The Atlas framework by Bosse et al. (2004) is a good example of this strategy. Moreover, several recent algorithms as Olson et al. (2006) GraphSLAM and Grisetti et al. (2007) TORO have proved an even improved performance optimizing such graphs, while the problem of data association often remains an open issue.

This paper deals with the problem of correspondence search between two different grid maps of limited size in which an initial estimation of the position between them is provided. We propose a novel minimization technique that computes a pose estimation that minimizes the discrepancy or negative overlap between such grid maps. Our algorithm differs in other existing approaches in the sense that it not only computes matching correspondences in which a positive association is found between the grid maps, but it is also able to deal with non matching correspondences, leading to a solution that provides a feasible physical position between the maps. Up to our knowledge this is the first algorithm that can handle positive data association and physically impossible configurations between grid maps in a unified way, which can be extremely useful for many of the above cited SLAM algorithms.

Section 2 presents the basics and required notation of robotic mapping with grid maps. Section 3 describes the cost function defined for establishing the discrepancy between overlapping grid maps, while Section 4 applies such cost function in a quasi Newton minimization algorithm. Section 5 provides experimental evidence that our algorithm provides a feasible and robust solution for handling overlapping grid maps, not only finding correspondences but also finding feasible physical configurations without correspondences. Our conclusions are summarized in Section 6.

## 2. ROBOTIC MAPPING

### 2.1 Grid maps

In grid mapping, the map $M$ is a rectangular regular grid divided into small size square cells that accumulates all the sensor information $z^{t}$ and robot poses $x^{t}$ up to time step $t$, representing in probabilistic terms the posterior $p\left(M \mid x^{t}, z^{t}\right)$, where the conditioning is known as mapping with known poses, i.e. the poses of the robot along its trajectory are assumed to be correct (typically being estimated by other means). To handle the high dimensionality of the problem, each cell occupancy $m_{\mathrm{c}}$ is supposed to be independent, so the posterior can be factorized as:

$$
\begin{equation*}
p\left(M \mid x^{t}, z^{t}\right)=\prod_{\mathbf{c}} p\left(m_{\mathbf{c}} \mid x^{t}, z^{t}\right) \tag{1}
\end{equation*}
$$

Where $\mathbf{c} \triangleq\left(\begin{array}{ll}i & j\end{array}\right)^{T} \in \mathbb{I}^{2}$ represents the natural indices of a cell, with $i, j$ taking all possible values in the rectangular grid so that $m_{\mathbf{c}} \in M$. The resolution of the grid is defined by the size of each square cell, which we will note as $\delta$.

Without loss of generality, the grid map is defined to be attached to a reference frame with axis X and Y aligned with $i, j$ indices respectively, and the origin displaced by an offset $\Delta \triangleq\left(\begin{array}{ll}\Delta_{x} & \Delta_{y}\end{array}\right)^{T} \in \mathbb{R}^{2}$. Thus, a scale function $\lambda$ that converts map indices into real positions (relative to the map frame) $\mathbf{v}=\left(\begin{array}{ll}x & y\end{array}\right)^{T} \in \mathbb{R}^{2}$ can be defined as:

$$
\begin{equation*}
\mathbf{v}=\lambda(\mathbf{c})=\mathbf{c} \times \delta-\Delta \tag{2}
\end{equation*}
$$

Conversely, the inverse relation can be defined (the brackets represent the floor() operation), to computes map indices corresponding to the cell in which a point real coordinates fall in:

$$
\begin{equation*}
\mathbf{c}=\lambda^{-1}(\mathbf{v})=\left\lfloor\frac{\Delta+\mathbf{v}}{\delta}\right\rfloor \tag{3}
\end{equation*}
$$

### 2.2 Relative transformations

The relative position and orientation (pose) of a reference frame $B$ respect another frame $A$ can be defined by a vector $\mathbf{r}_{A B}=(x, y, \theta)^{T}$ with translation $x, y$ and rotation $\theta$. Given a point $\mathbf{v}_{B}$ expressed in reference $B$, the computation of the coordinates of that point in frame $A$ is done by:

$$
\mathbf{v}_{A}=\mathrm{T}\left(\mathbf{r}_{A B}\right) \times \mathbf{v}_{B}=\left[\begin{array}{cc}
\cos \theta & -\sin \theta  \tag{4}\\
\sin \theta & \cos \theta
\end{array}\right] \mathbf{v}_{B}+\left[\begin{array}{l}
x \\
y
\end{array}\right]
$$

Where $\mathbf{T}\left(\mathbf{r}_{A B}\right)$ is a compact representation of the change of base operation.

Note that in the SLAM graph and tree approaches cited in the introduction, the nodes are commonly defined by the reference frames of the submaps, and edges are consequently roto-translations as established above. Even if the relative pose of two nodes is not defined in the graph by the corresponding edge, it can be easily computed by compounding relative transformations along the shortest path connecting those nodes.

## 3. DISCREPANCY BETWEEN GRIDS

The key idea of the work presented in this paper is that an objective measure of the difference or discrepancy between two (totally or partially) overlapping grid maps can be easily defined and computed, and subsequently used in a optimization algorithm in order to minimize such discrepancy. The discrepancy is seen as an error measure, a non negative number.

Three scenarios are possible. In the simplest case, there are no cells labelled as obstacles in a map that fall into non occupied cells (free space) in the other one. In such a case the discrepancy measure should be equal to zero, it is already at its minimum and there is nothing to minimize. In the more general case shown in Fig. 1, obstacle cells of a map will fall in free space of the other one, in a clear discrepancy under the common assumption of a static world. In the second possible scenario, there exists a relatively close to the initial position correspondence between the maps in which the discrepancy should be theoretically zero, and that should be found in the minimization procedure. The third scenario is encountered when there is discrepancy between two maps, but there exists no correspondence between them. As the relative pose is not physically possible, the minimization should ideally lead to the closest configuration with a more feasible discrepancy.

### 3.1 Discrepancy of overlapping grid maps

Let one map be denoted as $M_{A}$, in which the probability of a cell defined by its indices $\mathbf{c}_{A}$ in that map, has probability of occupancy $p_{A}\left(m_{\mathbf{c}_{A}}\right)$ in which the conditioning has been omitted for clarity. Similarly, $p_{B}\left(m_{\mathbf{c}_{B}}\right)$ defines the probability of occupancy of cell with indices $\mathbf{c}_{B}$ of map $M_{B}$.


Fig. 1. Two overlapping grid maps. Occupied cells are shown in dark, free space in white. Blue are non explored (unknown) areas. Map B is showed translucent, so the discrepancy (dark cells of B falling in free space of A and vice versa) becomes visible.

For each cell of a map it is necessary to compute the overlapping cell of the other one. For this purpose it is very convenient to define a function $f$ that maps the cell indices of map $M_{B}$ to overlapping cell indices of map $M_{A}$. Using (2), (3) and (4), such function can be defined as:

$$
\begin{equation*}
f\left(\mathbf{c}_{B}\right)=\lambda_{A}^{-1}\left(\mathrm{~T}\left(\mathbf{r}_{A B}\right) \times \lambda_{B}\left(\mathbf{c}_{B}\right)\right) \tag{5}
\end{equation*}
$$

With the use of (5), the error $e$ or discrepancy between maps $M_{A}$ and $M_{B}$ with relative position defined by $\mathrm{T}_{A B}$ could be computed as:

$$
e\left(M_{A}, M_{B}, \mathbf{r}_{A B}\right)=\sum_{\substack{\mathbf{c}_{B} \in M_{B} \\ \wedge f\left(\mathbf{c}_{B}\right) \in M_{A}}}\left|p_{A}\left(m_{f\left(\mathbf{c}_{B}\right)}\right)-p_{B}\left(m_{\mathbf{c}_{B}}\right)\right|
$$



Fig. 2. Discrepancy between two grid maps of the same corridor with different relative positions

When defining a cost function for minimization purposes, it is interesting to have a look at it first, in order to check its suitability. A grid map of 12 m width with a cell resolution of 0.025 m of a 6 meters wide straight corridor is used as the base map $M_{A}$ and exactly a copy of it is used as $M_{B}$ (Fig. 1).

The error is computed with (6) for many different displacements $x, y \in(-12,12)$ with $\theta$ equal to zero and is plotted in the Fig 2.

Figure 2 shows a deep flat valley with zero discrepancy for $y=0$. This setting obviously represents the perfect alignment of the X axis of both maps, and a perfect correspondence. The flat areas for values $y \notin[-66]$ correspond to translations along the Y axis larger than the corridor width, and so are feasible physical configurations with zero discrepancy. So far, the behaviour of the error function is adequate. The problem is that many common optimization algorithms assume that the function can be locally approximated by a quadratic function. This is absolutely false for our function, resembling two triangular prisms with sharp edges. The problem of such edges is that a relative pose between the maps can be very close to the minimum and desired solution, as close to the edges of the prisms, for example for values $x=0, y=1$. Moving just 1 meter along Y leads to zero discrepancy (a close and desired solution). Instead, any gradient based minimization approach will evolve along the slope, displacing $y$ more than 12 meters. It seems reasonable trying to find a smoother error function.

### 3.2 Blurring the submaps

The first idea could be to compute the full error function values, then filter or smooth it. It has to be taken into account that the evaluation of (6) is a costly operation, because it has to handle thousands of cells. If an excessive number of evaluations were required, the computational burden would render our approach useless. Minimization algorithms do not require the evaluation of the full search space, but only at certain (hopefully few) points. So instead of trying to modify the error function, they are applied to a modified map. From the original grid map, a new one is computed applying a standard blurring algorithm that introduces a gradient of grey from black (occupied) to white (free) cells. This is a quite fast operation that has to be computed only once for each grid map.


Fig. 3. Original (left) and blurred (right) grid maps.

Applying exactly the same settings as in Fig 2, the error function is computed and plotted in Fig. 4. It can be easily appreciated that it has been gently smoothed, thus being more suitable for optimization purposes.


Fig. 4. Discrepancy between two grid maps of the same corridor with different relative positions, using the blurred maps.

Please recall that although the original grid maps are shown in the remaining figures, all the evaluation of discrepancies is carried out using the computed blurred versions.

### 3.3 The cost function

There is still a last problem with the discrepancy function as defined so far. The flat areas are a problem for most optimization algorithms. Especially in our case, a large slope of the discrepancy suddenly ends in a flat area, which can easily produce an overshot in the solution search, finding a local (and most importantly, global) minimum very far away from the initial relative pose.

The problem is that the distance to the initial pose $\mathbf{r}_{A B}^{0}$ has not been taken into account. Put verbally, a solution relatively close to the initial position is desired. This is important if the overall SLAM framework is considered, because the initial relative pose will typically be the best currently available estimation (the minimum energy one), and consequently solutions closer to the initial pose will have a larger likelihood.

Thus, a distance measure between two reference frames has to be established. Note that Euclidean distance between the frames origins is not an option, as it doesn't take into account the rotation. We propose as in to use a weighted distance that conveniently scales with a factor $L$ the effect of rotations compared with translations:

$$
\begin{equation*}
\left\|\mathbf{r}_{A B}\right\| \triangleq \sqrt{x^{2}+y^{2}+L^{2} \theta^{2}} \tag{7}
\end{equation*}
$$

Our final cost function $h$ used in the minimization described in next section is defined as a weighted sum of (6) and (7):

$$
\begin{align*}
& h\left(M_{A}, M_{B}, \mathbf{r}_{A B}, \mathbf{r}_{A B}^{0}\right)= \\
& \quad=e\left(M_{A}, M_{B}, \mathbf{r}_{A B}\right)+K\left\|\mathbf{r}_{A B}-\mathbf{r}_{A B}^{0}\right\| \tag{8}
\end{align*}
$$



Fig. 5. Cost function h evaluation between two grid maps of the same corridor with different relative positions, using blurred maps and taking into account the distance to the initial pose.

It can be seen that the second term of (8) produces a quadratic that makes the cost higher as the pose gets further from the origin. The final result is a smooth cost function that can be locally approximated by quadratics and with clear minima that an adequate minimization algorithm should find.

## 4. DISCREPANCY MINIMIZATION

A quasi Newton minimization algorithm is employed, so the explicit computation of the Hessian matrix is not required. The numeric computation of the Hessian matrix requires many (8) evaluations and, as explained above, this is undesirable as such evaluations are relatively costly. Put mathematically, the problem can be defined as:

$$
\begin{equation*}
\mathbf{r}_{A B}^{*}=\underset{\mathbf{r}_{A B}}{\arg \min } h\left(M_{A}, M_{B}, \mathbf{r}_{A B}, \mathbf{r}_{A B}^{0}\right) \tag{9}
\end{equation*}
$$

The Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm (Nocedal, 2006) is well known to perform well in many situations. We have employed the open source implementation of Bochkanov, (2010). At each iteration the value and the gradient of the function have to be computed. The four points formula is used as shown in (10) for $x(y$ and $\theta$ are analogous). $M_{A}, M_{B}, y, \theta, \mathbf{r}_{A B}^{0}$ parameters are omitted for clarity, and $\boldsymbol{\varepsilon}=\left(\begin{array}{lll}\varepsilon & 0 & 0\end{array}\right)^{T}$ :

$$
\begin{align*}
\frac{\partial h\left(\mathbf{r}_{A B}\right)}{\partial x}=\frac{1}{12 \varepsilon} & \left(h\left(\mathbf{r}_{A B}-2 \boldsymbol{\varepsilon}\right)-8 h\left(\mathbf{r}_{A B}-\boldsymbol{\varepsilon}\right)\right.  \tag{10}\\
& \left.+8 h\left(\mathbf{r}_{A B}+\boldsymbol{\varepsilon}\right)-h\left(\mathbf{r}_{A B}+2 \boldsymbol{\varepsilon}\right)\right)
\end{align*}
$$

As proved in next section, such cost function and minimization algorithm provides excellent results. Further analysis of the discrepancy defined in (6), shows that useful information can be extracted from the minimum found. Consider the flat valley of Fig. 4. Displacements along the X
axis provide no change of discrepancy, but displacements along the Y axis quickly increase the error. Thus, the curvature of the discrepancy function is a measure of the information provided by the new relative pose computed by the minimization algorithm:

$$
\begin{equation*}
\operatorname{cov}\left(\mathbf{r}_{A B}^{*}\right) \propto(\mathbf{H}(e))^{-1} \tag{11}
\end{equation*}
$$

When the algorithm arrives at a minimum, the Hessian matrix
$\mathbf{H}$ is computed numerically at that point. Note that this evaluation is done only once, at the end of the minimization. Next equation shows the computation of one diagonal and one off diagonal component, in which $M_{A}, M_{B}$ parameters have been omitted for clarity

$$
\begin{align*}
& \boldsymbol{\varepsilon}_{x}=\left(\begin{array}{lll}
\varepsilon & 0 & 0
\end{array}\right)^{T}, \boldsymbol{\varepsilon}_{y}=\left(\begin{array}{lll}
0 & \varepsilon & 0
\end{array}\right)^{T} . \\
& \begin{aligned}
\frac{\partial^{2} e\left(\mathbf{r}_{A B}\right)}{\partial x^{2}}= & \frac{1}{12 \varepsilon^{2}}\left(-e\left(\mathbf{r}_{A B}-2 \boldsymbol{\varepsilon}_{x}\right)+16 e\left(\mathbf{r}_{A B}-\boldsymbol{\varepsilon}_{x}\right)\right. \\
& \left.-30 e\left(\mathbf{r}_{A B}\right)+16 e\left(\mathbf{r}_{A B}+\boldsymbol{\varepsilon}_{x}\right)-e\left(\mathbf{r}_{A B}+2 \boldsymbol{\varepsilon}_{x}\right)\right)
\end{aligned} \\
& \begin{array}{r}
\frac{\partial^{2} e\left(\mathbf{r}_{A B}\right)}{\partial x \partial y}= \\
\frac{1}{4 \varepsilon^{2}}\left(e\left(\mathbf{r}_{A B}-\boldsymbol{\varepsilon}_{x}-\boldsymbol{\varepsilon}_{y}\right)-e\left(\mathbf{r}_{A B}-\boldsymbol{\varepsilon}_{x}+\boldsymbol{\varepsilon}_{y}\right)\right. \\
\\
\left.-e\left(\mathbf{r}_{A B}+\boldsymbol{\varepsilon}_{x}-\boldsymbol{\varepsilon}_{y}\right)+e\left(\mathbf{r}_{A B}+\boldsymbol{\varepsilon}_{x}+\boldsymbol{\varepsilon}_{y}\right)\right)
\end{array}
\end{align*}
$$

## 5. EXPERIMENTS

The first experiment analyzes the ability of the algorithm to find a positive correspondence solution. A grid map 12 meters square, with a resolution of 0.025 m is used. It corresponds to an environment with a straight corridor 6 meters wide. The other map is a copy of it, but it is initially located in a random pose in the vicinity of the origin, as shown in Fig. 6. The minimization procedure should get a solution in which the X axes of the maps are aligned, but not necessarily aligning the Y axes. This is an obvious consequence of the environment symmetry. Basically any matching algorithm should provide corrections only in the transversal and angular directions, as no information is available along the corridor axis, i.e. the discrepancy is the same for all configurations with the X axes aligned, irrespective of their longitudinal relative position. Our discrepancy definition (6) accounts for this symmetry as shown in Fig. 6, while the cost function (8) takes into account the relative distance, thus finding the closest pose along the corridor axis. Furthermore, this symmetry is captured by the solution covariance, also represented in Fig. 6.

This setting is repeated 100 times, with random initial poses ranging in different intervals, as shown in Table 1. The percentage of solutions that fall into an interval is represented. Note that such solution interval is defined only in terms of $y$ and $\theta$, because of the symmetry of the corridor along X .


Fig. 6. Initial setting (left) and final solution (right) finding a positive correspondence with the maps correctly aligned. The covariance ellipsoid (just XY components) is shown in cyan. It can be easily seen that a very low covariance is obtained for the Y direction, while the information along the X axis (corridor axis) is extremely low (zero in the perfect case), and so a very large uncertainty along X is obtained.

Table 1. Accuracy, matching case

|  | Accuracy (m,rad) |  |
| :---: | :---: | :---: |
| Noise $(m, r a d)$ | $\|y\|<0.025,\|\theta\|<0.045$ | $\|y\|<0.05,\|\theta\|<0.09$ |
| $x, y, \theta \in \pm 0.5$ | $99 \%$ | $1 \%$ |
| $x, y \in \pm 1.5$, <br> $\theta \in \pm 0.38$ | $98 \%$ | $0 \%$ |

The average number of L-BFGS iterations is 35 , with an average computation time for the whole minimization procedure of 6 seconds (Core 2 Duo @ 2 Ghz laptop).


Fig. 7. Initial setting (left) and final solution (right) finding a relative pose corresponding to a physically feasible configuration, although with a slightly poorer alignment and a displacement along X axis larger than expected

It is important to highlight that the few minimizations leading to solutions out of the given intervals, does not necessarily mean an incorrect or unfeasible relative pose between the grid maps. Fig. 7 shows one of those cases.

One of the contributions of this work is that the presented algorithm is able to handle any overlap between grid maps, even if a correspondence does not exist between them. Consider the case of Fig. 8, in which the same grid maps as above have been used, but with a larger initial relative position. In this case, it is more likely that the real configuration of the environment is one with two parallel corridors, as shown in Fig. 8 (right). Table 2 summarizes the average results for 100 runs.


Fig. 8. Initial setting (left) and final solution (right) finding the closest relative pose corresponding to a physically feasible configuration in which the discrepancy in the overlap is minimized.

Table 2. Accuracy, parallel corridors case

|  | Accuracy (m,rad) |  |  |
| :---: | :---: | :---: | :---: |
| Noise <br> $($ m,rad $)$ | $\|y\|<0.025$ | $\|y\|<0.05$ | $\|y\|<0.2$ |
| $\|\theta\|<0.045$ | $\|\theta\|<0.09$ | $\|\theta\|<0.16$ |  |
| $x \in \pm 1.5$ <br> $y \in 5 \pm 1.5$ <br> $\theta \in \pm 0.38$ | $62 \%$ | $73 \%$ | $93 \%$ |

In this case our algorithm adequately manages the symmetry of the corridor, as can be seen in the covariance ellipsoid. The covariance in the Y direction is larger than in Fig. 6 because the configuration has less information; it is less constrained than that shown in Fig. 8.

The average number of iterations is 23 , with an average processing time of 3.6 seconds. These results are coherent. As the maps are repelled while minimizing the discrepancy, less information is available than in the matching case where the overlap is much larger, so the alignment of corridors is more difficult. Despite the lower accuracy the robustness of the method is as good as in the matching case. Consider that not being able to achieve an expected alignment does not mean that the obtained relative position is wrong. Fig. 9 shows an example. Although the final solution does not present a perfect parallelism, it is a physically feasible configuration, and so it is more correct than the initial one.


Fig. 9. Initial setting (left) and final solution (right) finding a relative pose corresponding to a physically feasible configuration, although with a slightly poorer alignment than expected

## 6. CONCLUSIONS

This paper has presented a novel minimization algorithm that can compute a physically feasible relative pose between two overlapping grid maps given an initial estimation, by minimizing the discrepancy between them. Experimental results have proved that the approach is able to robustly and accurately find correspondences, but also (and this is an important contribution) to handle negative information under a unified approach. In either case the resulting pose and covariance are of extreme interest for SLAM algorithms. Although the experiments presented are based on simulations for simplicity and explanatory reasons, our preliminary tests show a similar performance for real data. Further work includes a deeper study of the typical discrepancy values, number of local minima and the division of the configuration space in areas corresponding to each minimum, in order to identify and model the full solution space.

## ACKNOWLEDGMENT

This work was supported by Spanish Ministry of Science and Technology (project Robonauta: DPI2007-66846-C02-01).

## REFERENCES

Bochkanov, S. (2010). ALGLIB software library, L-BFGS C++ implementation. http://www.alglib.net/

Bosse, M., Newman, P., Leonard, J. and Teller, S. (2004). Simultaneous Localization and Map Building in LargeScale Cyclic Environments Using the Atlas Framework. International Journal of Robotic Research. Vol. 23. Issue 12. pp 1113-1140.

Elfes. A. (1987). Sonar-based real-world mapping and navigation. IEEE Journal on Robotics and Automation. Vol. 3 N. 3. pp. 249-265.
Grisetti, G., Stachniss C., Grzonka S., and Burgard W. (2007) A Tree Parameterization for Efficiently Computing Maximum Likelihood Maps using Gradient Descent. Robotics: Science and Systems (RSS), Atlanta, USA.

Hahnel D, Burgard W, Fox D., Thrun S. (2003) An efficient FastSLAM algorithm for generating maps of large-scale cyclic environments from raw laser range measurements. IEEE/RSJ IROS'03.

Moravec H., Elfes A. (1985) High resolution maps from wide angle sonar. IEEE International Conference on Robotics and Automation. Vol. 2. pp. 116-121. 1985.
Nocedal, Jorge; Wright, Stephen J. (2006), Numerical Optimization (2nd ed.), Berlin, New York: SpringerVerlag, ISBN 978-0-387-30303-1
Olson, E. and Leonard J. and Teller S. (2006). Fast Iterative Optimization of Pose Graphs with Poor Initial Estimates, IEEE ICRA’06, pages 2262-2269

Thrun. S. (2002). Robotic mapping: A survey In G. Lakemeyer and B. Nebel, Exploring Artificial Intelligence in the New Millenium. Morgan Kaufmann.

