Optimizing range finder sensor network coverage in indoor environment

J. Ferreira*, A. Vale**, R. Ventura*

*Institute for Systems and Robotics/Instituto Superior Técnico, Av. Rovisco Pais,1; 1049-001 Lisboa, (joao.teles.ferreira@ist.utl.pt, yoda@isr.ist.utl.pt)
**Instituto de Plasmas e Fusão Nuclear/Instituto Superior Técnico, Av. Rovisco Pais,1; 1049-001 Lisboa (avale@ipht.ist.utl.pt)

Abstract: This document states a method to optimize the coverage of a range finder multisensor network on an indoor environment. The problem arose from the need of a localization system for autonomous mobile robots, integrated with the building. The common approach for a localization system consists on sensors in the vehicle itself. In this case the vehicle is only an actuator and the sensory part is installed on the building. The localization system performance depends on the sensor placement, to enhance it, the sensor network must maximize the covered area. To solve the problem, an algorithm that maximizes the coverage areas is proposed. Given a scenario map, obtains the optimal coverage area multisensor placement. This paper presents also some results and proposals to future approaches to the problem.

Keywords: Polygons, Multisensor integration, Range Finder Sensor

1. INTRODUCTION

Guiding autonomous vehicles to navigate to a given goal position requires a well designed positioning system. Inside a building with narrow spaces, maneuvering a vehicle is a hard problem. So the position and orientation estimation must be accurate and precise.

In general, vehicle estimates his own position and orientation using on board sensors. The proposed solution uses building installed sensors, while the vehicle is sensor free. This choice is motivated from the need to estimate positioning on the ITER (International Thermonuclear Experimental Reactor) buildings. On this platform the vehicle transports radioactive products, being the main radiation source. Radiation can decrease rapidly the lifetime of onboard sensors, so the best option is to install them on the building where the radiation only passes by. Using only these sensors, the poses of multiple vehicles can be estimated simultaneously, with the same sensor network, which would be impossible for on board sensors.

To implement this location system, it is crucial that the sensor network cover the entire scenario, avoiding occlusion situations. Indoor environments have, typically, many obstacles and occlusions, only one sensor can never cover the entire area. To have a complete coverage of the environment it is required a group of sensors with different configurations. Optimizing these configurations is crucial to the system’s performance.

This paper proposes a method for optimizing sensor network placement, maximizing the coverage, with the goal of minimizing the amount of sensors to be installed.

The sensor coverage in an indoor environment is a well known problem, since the “art gallery problem” stated by V. Chvátal (1975). This problem inquires how many observation points are necessary to cover an entire area with a given number of walls. This is a NP-problem as proved by Aggarwal (1984). Any solution to these problems is very fast to verify but there is no fast solution known. The difficulties in optimization on a visibility problem like this are well understood, and many approximate solutions have been proposed, applied usually in visual sensors, like cameras in surveillance systems. A. Mittal, L.S. Davis (2002) and (2004) developed a method to optimize coverage areas and enhance performance of visual sensors network. The improvement of visual sensor network coverage, like surveillance cameras (Yabuta and Kitazawa (2008)) have similar optimizing difficulties, the main constraints are the same, the occlusions in indoor environments.

The solution proposed uses visual sensors installed on the building, these sensors measure the surrounding environment and, combining their measures, estimate the location of a moving object, like a vehicle. In the particular application on ITER scenarios, the sensors proposed are range finder lasers.

This document is organized as follows. Section 1 introduces the solution proposed with multiple range finder sensors. Section 2 presents an algorithm to compute the visibility polygon for a single sensor and how to use these results to optimize sensor placement. Section 3 proposes an optimization process to choose multiple sensor placements on a map. Section 4 presents some results and guide lines to choose the best configuration obtained. Section 5 concludes the paper and presents directions for further work.
2. SINGLE SENSOR PLACEMENT

The goal in this section is to obtain a Visibility Polygon (VP), given the map and the sensor position and orientation. VP represents the total area seen from sensor taking in consideration the occlusions, the range of the sensor and his Field of View (FoV). The polygon is described by the coordinates of each vertex and by the order by which they are connected. For a single sensor, VP is always a star-shaped polygon, as defined by Preparata and Shamos (1985), meaning that, from, at least, one point inside the VP all VP can be seen. With one VP for each sensor configuration it is possible to optimize the coverage for various sensors.

Scenario map (1) and sensor information (2) are required to compute VP.

\[ MP = \begin{bmatrix} x_1 & y_1 & r_1 \\ \vdots & \vdots & \vdots \\ x_N & y_N & r_N \end{bmatrix}, \quad ML = \begin{bmatrix} i_1 & f_1 & v_1 \\ \vdots & \vdots & \vdots \\ i_M & f_M & v_M \end{bmatrix} \]

(1)

\[ s = [t \ \theta \ \beta \ \phi]^T \]

(2)

Map is basically a polygon with holes; the exterior polygon is a sequence of points in counter clock wise (CCW) it represents the outer walls and corners. Holes represent obstacles, smaller polygons inside the exterior one. Each one is a sequence of point in clock wise (CW).

The map is built from a set of N points MP, and M lines, ML. Each point is represented by his coordinates x and y and a flag r. A point is always a connection between two lines, if the greater angle between these lines is an inside angle of the map polygon r is True, if not r is False. ML is the set of walls on the map, each one with an initial and a final point, i and f are the indexes of those points in the set MP, respectively. v is a flag stating if the sensor can be installed on that wall or not (Fig. 1). The sensor state vector s, is composed by \( t \in [0,1] \), a parameter of sensor position along the wall, \( \theta \) is the sensor orientation, \( \beta \) is the wall where the sensor is installed and \( \phi \) is the sensor’s Field of View (Fig. 2a). With \( t=0 \) the sensor is installed in the initial point of wall \( \beta \), with \( t=1 \) in the final point. Sensor movement along a wall from initial to final point is CCW in outer walls and CW in inner walls.

Fig. 1. Example of Map representation

VP for sensor state s is represented by an array Poly(s) (3) with the vertexes in CCW. The order of the points in the vector is important to represent the correct polygon.

\[ Poly(s) = \begin{bmatrix} x_1 & \ldots & x_k \end{bmatrix} \]

(3)

Fig. 2. Sensor state representation (a) VP – Example with \( \phi=180^\circ \) (b)

The algorithm implemented to obtain Poly(s) works with the points of the map and tests all the occlusions. Initially all points are organized CCW taking in consideration the sensor orientation. This is useful to exclude points outside the FoV and to have an organized set to build Poly array.

From the sequence of points, the algorithm checks if there are intersections and compute the side of each point. Corners with no occlusions enter the Poly(s) and, since they are already organized there is no need to further operations. Fig. 3 shows the basic algorithm.

1. Initialize sensor pose s;
2. Organizes map points CCW way;
3. Excludes map points outside the FoV;
4. Initialize an empty Poly(s) array;
5. Include, in Poly(s), the polygon limit vertexes limited by the FoV or walls;
6. For each point:
   a. Calculates SIDE of point;
   b. For each map line:
      i. Calculates INTERSECTION between segment (from sensor to point) and map line;
      ii. If the line occludes the point, break cycle, if not, registers the closest intersection;
   c. If the point is on a side, (side=1,-1) the corner and the closest intersection (node) enter Poly array, if the point is in the middle (side=0), only himself enters the array;
7. Exports Poly(s) array.

Fig. 3. Algorithm

These are the major steps of the algorithm that returns a polygon ordered in CCW. The worst case scenario complexity is \( O(N*M) \), this depends on the FoV and even on sensor position.

The main functions mentioned in the algorithm are SIDE (Fig. 4a) and INTERSECTION (Fig. 4b), they use simple geometric tools explained in this section.
SIDE – Knowing that each corner point (corner) is part of an obstacle, and this obstacle have three possible configurations respecting the direction of observation. It returns a value for each configuration (4). This value is used to know if the corner is the only point to put on the Poly(s) array or if it should be obtained a new point behind him (node). The value is also useful to know in which order (Fig.4a arrows) to put node and corner on Poly(s).

\[
S\text{IDE} = \begin{cases} 
-1, & \text{right \ Fig.4a) B} \\
0, & \text{middle \ Fig.4a) C} \\
1, & \text{left \ Fig.4a) A} 
\end{cases} 
\]  

INTERSECTION – This function is the core of the algorithm, it makes the intersection between a segment that goes from sensor position to map point (corner), and map lines. In the end it returns two values that parameterize the intersection (5). Evaluating the combination between these two values it is possible to know the point of intersection and if the map line occludes the point. Occlusions happen (Fig. 4b) when \(\lambda_1\) and \(\lambda_2\) are both between 0 and 1.

\[
\begin{align*}
\lambda_1 &= \frac{f_x - i_x}{f_x - i_x} - s_x - cor_x^{-1}s_x - i_x \\
\lambda_2 &= \frac{s_y - i_y}{s_y - i_y} - s_x - cor_y^{-1}s_x - i_x 
\end{align*}
\]  

(\(\lambda_1,\lambda_2\)) are the intersection parameters, \((f_x,f_y)\) and \((i_x,i_y)\) are the coordinates of the line final and initial points, respectively. \((s_x,s_y)\) are coordinates of the sensor position and \((cor_x,cor_y)\) are coordinates of the corner point.

The results for various sensor poses are obtained making a function \(\text{Pol}:S \rightarrow \text{VPS}\), that, given a set of sensor states \(S = \{s_1, s_2, ..., s_{\text{VPS}}\}\), calls the algorithm described for each state in \(S\). The result is a set of \(\text{VP}\), \(\text{VPS} = \{\text{Poly}(s_1), \text{Poly}(s_2), ..., \text{Poly}(s_{\text{VPS}})\}\).

The state set \(S\), consists on various positions and orientations of sensor on the wall of the map. Increasing \(t\) from 0 to 1 for each value of \(\beta\) changes the sensor position. Rotations are changes of \(\phi\) for each position, conditioned by the FoV. If \(\phi=180^\circ\) the orientation is fixed along the walls, changing only in corner points.

On this implementation \(S\) is a discrete set, a discrete number, \(\text{tag} \in \{1,2, ..., W\}\) is attributed to each sensor state. Lowest \(\text{tag}=1\) corresponds to sensor state \(s_1\), \(\text{tag}=2\) to \(s_2\), and so on. These \(\text{tags}\) help to identify the sensor state as well as the respective \(\text{VP}\).

Results shown assume a FoV of 180°, that’s why the orientation is always normal to the walls. The discretization of \(S\) is with translation steps of 0.2m and rotation steps on the corners of \(\pi/80\) rad.

A basic situation, on a square map, with one obstacle and one sensor (Fig. 5a), illustrates the procedure obtaining coverage areas. The sensor movement, shown by the blue arrows, is CCW around the exterior walls (region A) and CW in interior walls (region B). The area graph (Fig. 5b) showing the coverage in percentage marks clearly the map’s symmetry. The situation shown (tag=182) is a maximum, but due to the symmetry is not the only one, many local maximum, and many narrow valleys are the main characteristic of this problem, which turns the optimization a very hard task.

Valleys are directly connected to abrupt changes on area values, this happens normally during rotations. If the point of rotation is very central (Fig. 6a) the area gained (A3) balances the area lost (A1). Otherwise, if the point of rotation is very close to a wall and very far from the other (Fig. 6b), the area lost (A1) and gained (A3) are not equivalent creating an abrupt change on the total area observed. These abrupt changes create a narrow valley and climbs.

Moving to more complex scenarios, the ITER map (Fig. 7a) and a general warehouse map (Fig. 7b) are analyzed. Both have approximated dimensions but ITER has less clear area than the warehouse.

The coverage area graph for ITER map (Fig. 8) shows three regions, each one corresponds to:

- Region A (Fig. 7a dashed arrows) – Outer walls;
- Region B (Fig. 7a solid arrows) – Pillar walls;
- Region C (Fig. 7a dotted arrows) – Inner walls.

Regions B and C have some discontinuities due to some jumps on the sensor state, in B, from pillar to pillar and in C, from wall to wall. Observing Fig. 8 the better mean coverage values are in outer walls, this means that the sensors that contribute more to the system will be in these walls.
For warehouse map only external walls are considered to place the sensor. The maximum shown on each coverage area graph is shown also on each map (Fig. 7).

In both graphs it can be found similar regions where the behavior of the curve is almost the same. In ITER graph around tag=1200 and tag=2500, there are two similar regions. These are made when the sensor position is on the map’s side walls, creating similar curves due to symmetry on the map. The same happens in Warehouse example, when the sensor travels the bottom and top walls (around tag=200 and tag=1700 on the area graph).

The coverage area graphs, to the three scenarios, present some symmetry. In general case, human constructions have symmetry, so this is a typical behavior for the coverage area curves.

3. MULTI SENSOR PLACEMENT

To create one localization system embedded on the building, the area where the autonomous vehicles can travel must be observed by, at least, one sensor. Generally, only one sensor does not cover the entire scenario; there are occlusions due to construction and obstacles.

To position various sensors the following criteria were taken into consideration:

- Coverage Area – Total area covered, corresponding to the union of the VP of each sensor;
- Redundancy – Area covered by at least two sensors;

The current Performance Measure Function, \( F \) in (6) considers only the Coverage Area. So the greater the area covered by the sensors the best. In this problem the optimization consists on maximizing \( F \), in (7).

\[
F(s_1, ..., s_L) = \text{Area}[\text{Pol}(s_1) \cup ... \cup \text{Pol}(s_L)]
\]

\[
F'(L) = \max_{s_1, ..., s_L} F(s_1, ..., s_L)
\]

\[
S^*(L) = \arg\max_{s_1, ..., s_L} F(s_1, ..., s_L)
\]

Where \( s \in S \) represents a sensor state, \( L \) is the number of sensors, \( \text{Area} \) is a function that returns the area of a polygon.

\( F'(L) \) is the optimal value of area covered by a network of \( L \) sensors and \( S^*(L) \) the respective set of sensors.

In this section, the \( \text{VPS} \) (View Polygon Set) obtained earlier is useful to calculate the \( F \) value for each combination of sensor poses.

As said the optimization on this problem consists on the maximization of \( F \). The main problem to this maximization is the fact that the function has many narrow valleys. The graph presented in Fig. 8 and Fig. 9 show the complexity of finding a global maximum for \( L=1 \). As \( L \) increases the complexity grows exponentially, \( F \) is a nonlinear, nonsmooth function and so, it is proposed a Monte Carlo Method to solve the optimization problem.

There are many algorithms that can suite this problem but the choice was the Simulated Annealing (SA) as described by D. Bertsimas and J. Tsitsiklis (1993). Having the possibility of evolving to worst solutions, the algorithm avoids local maximums and narrow valleys. In this case this propriety is important due to the number of valleys and local maximums. SA algorithm is also simple to implement and computational efficient.

This algorithm produces results efficiently, because it only evaluates a solution per iteration. The results are good to a certain degree of complexity, but with many sensors involved, the convergence to a maximum coverage area is slow, due to the evaluation of \( F \). With many sensors the number of local maximum increase dramatically so the chances of the algorithm being trapped in a non-global maximum increases, decreasing the confidence in the result.

In the end, the number of sensors, \( L \), must suite each situation’s needs. The coverage and redundancy areas must be maximized to enhance localization system’s performance, both in observed area as in accuracy. This is achieved by maximizing the number of sensors. Ideally an infinite number guarantees a total coverage and infinite redundancy. On the other hand, \( L \) must be minimized has well, due to system’s cost and processing capability. An infinite number of measures could become untreatable. A balance between this to factors is not trivial. The results from the optimization process are essential to \( L \) choice. As expected, as \( L \) increases, the incremental gain \( g(L) = F'(L) - F'(L - 1) \) decreases until it is zero. This corresponds to saturation on the \( F \) i.e. the coverage area is now the total map area. More sensors only bring redundancy to the system.

4. RESULTS

Running the optimization process for ITER tokamak scenario it was possible to obtain several coverage areas, their values, \( F'(L) \) are registered in Table 1. It is possible to note that the total area covered increases with \( L \) (number of sensors), but the gain \( g(L) \) (explained in section 3) decreases. This behavior can justify the choice of \( L \).
The values in Table 1 show the coverage area for more than one sensor. This is a measure of redundancy, and should be maximized as well. As \( F \) neglects the redundancy, the results only show that redundancy tends to increase with the coverage and with the number of sensors, as in Fig. 11.

<table>
<thead>
<tr>
<th>( L )</th>
<th>( K=1 )</th>
<th>( K=2 )</th>
<th>( K=3 )</th>
<th>( K=4 )</th>
<th>( K=5 )</th>
<th>( K=6 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>41.35%</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>78.43%</td>
<td>0%</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>92.43%</td>
<td>12%</td>
<td>0%</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>4</td>
<td>96.67%</td>
<td>41%</td>
<td>3%</td>
<td>0%</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>98.23%</td>
<td>49%</td>
<td>8%</td>
<td>0%</td>
<td>0%</td>
<td>-</td>
</tr>
<tr>
<td>6</td>
<td>99.55%</td>
<td>65%</td>
<td>12%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>7</td>
<td>99.70%</td>
<td>70%</td>
<td>22%</td>
<td>1%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>8</td>
<td>99.93%</td>
<td>73%</td>
<td>21%</td>
<td>4%</td>
<td>1%</td>
<td>0%</td>
</tr>
<tr>
<td>9</td>
<td>99.99%</td>
<td>86%</td>
<td>28%</td>
<td>3%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>10</td>
<td>100.00%</td>
<td>92%</td>
<td>51%</td>
<td>10%</td>
<td>1%</td>
<td>0%</td>
</tr>
</tbody>
</table>

Table 1. ITER: Coverage Percentage

Fig. 10 presents two solutions computed in the optimization algorithm. These are the configurations for maximum coverage areas, \( S'(2) \) and \( S'(3) \).

The same exercise applied to a typical warehouse scenario yields the optimal configurations shown in Fig. 12a and Fig. 12b for \( S'(2) \) and \( S'(3) \), respectively. The coverage areas (Table 2) have the same behavior, although with different values. Coverage and redundancy grow with the number of sensors until the coverage saturates (Fig. 13). By comparison between Fig. 11 and Fig. 13, redundancy is always lower in the Warehouse building. Warehouse has bigger disjoint obstacles, so there are more occlusions and the areas common to more than one sensor are smaller. This explains the redundancy reduction.
This paper presented a method to find optimal poses for several LIDAR sensors. These poses take into consideration the total coverage area i.e. the total area observed by the sensor network. As seen from the results, a finite number of sensors are enough to cover the entire area. The main idea to retain is that, for typical scenarios, the behavior of the coverage area is similar and that the redundancy of the system grows with the area covered. An open issue is redundancy optimization the effect of occlusions caused by vehicles to other vehicles; this topic should be addressed in a future work.

The optimization process is only the first step to the main objective. The sensor network main task is to estimate the position and orientation of several vehicles inside the scenario from several measures and movement model. This estimation can be done using Kalman Filters, Bayesian Filtering or particle filters.

Future work in this area includes the realization of a better optimization algorithm, which includes more variables, like redundancy or areas of interest. The redundancy can be measured by the area covered by two or more sensors and the areas of interest can be included weighting the covered areas. The positioning system with the optimized sensor network should be addressed in future work as well. As the localization system is developed becomes more clear which optimization criteria should be optimized and their importance in the process.

ACKNOWLEDGEMENTS

This work is supported by Instituto de Plasmas e Fusão Nuclear – Associate Laboratory, Instituto de Sistemas e Robótica – Associate Laboratory and by FCT under Ciência 2007 Programme.

REFERENCES


Dimitris Bertsimas and John Tsitsiklis, (1993), Simulated Annealing, Statistical Science, Vol.8 No. 1


U. Murat Erdem and Stan Sclaroff,(2006), Automated camera layout to satisfy task-specific and floor plan-specific coverage requirements


5. CONCLUSIONS

This paper presented a method to find optimal poses for several LIDAR sensors. These poses take into consideration the total coverage area i.e. the total area observed by the sensor network. As seen from the results, a finite number of sensors are enough to cover the entire area. The main idea to retain is that, for typical scenarios, the behavior of the coverage area is similar and that the redundancy of the system grows with the area covered. An open issue is redundancy optimization the effect of occlusions caused by vehicles to other vehicles; this topic should be addressed in a future work.

The optimization process is only the first step to the main objective. The sensor network main task is to estimate the position and orientation of several vehicles inside the scenario from several measures and movement model. This estimation can be done using Kalman Filters, Bayesian Filtering or particle filters.

Future work in this area includes the realization of a better optimization algorithm, which includes more variables, like redundancy or areas of interest. The redundancy can be measured by the area covered by two or more sensors and the areas of interest can be included weighting the covered areas. The positioning system with the optimized sensor network should be addressed in future work as well. As the localization system is developed becomes more clear which optimization criteria should be optimized and their importance in the process.

ACKNOWLEDGEMENTS

This work is supported by Instituto de Plasmas e Fusão Nuclear – Associate Laboratory, Instituto de Sistemas e Robótica – Associate Laboratory and by FCT under Ciência 2007 Programme.

REFERENCES


Dimitris Bertsimas and John Tsitsiklis, (1993), Simulated Annealing, Statistical Science, Vol.8 No. 1


U. Murat Erdem and Stan Sclaroff,(2006), Automated camera layout to satisfy task-specific and floor plan-specific coverage requirements
