Proceedings of the 2003 IEEE/RSJ
Intl. Conference on Intelligent Robots and Systems
Las Vegas, Nevada · October 2003

BVP-Exploration: further improvements

Edson P. e Silva Jr
Centro Universitário La Salle
Av. Victor Barreto, 2288, 92010-000
Canoas, RS - Brazil
prestes@inf.ufrgs.br

Marcelo Trevisan[1], Marco A. P. Idiart[1], Paulo M. Engel[2]
[1]Instituto de Física, P.O. Box 15051,
[2]Instituto de Informática, P.O. Box 15064,
Universidade Federal do Rio Grande do Sul
91501-970 Porto Alegre, RS - Brazil
tmarcelo@if.ufrgs.br, idiart@if.ufrgs.br,
engel@inf.ufrgs.br

***Abstract*— We propose some techniques to improve the exploration method based on BVP and demonstrate that, at least, for exploration in two dimensional space, we can reduce enormously the computation cost of the harmonic function method for exploration without losing the important properties of the method. Switching between local and global potential calculation and using an adaptive activation window result in a computationally efficient and refined mapping process.**

I. INTRODUCTION

Exploratory behavior is essential to autonomy. The ability to learn maps increases the flexibility and usefulness of mobile robots in several situations, specially where human supervision is restricted or limited.

Exploratory navigation consists of determining by sensory detection all obstacles and goals of a given environment. The robot has to travel around all possible corners collecting information about localization and sizes. It has to recognize objects, landmarks and free-space. Finally, based on the available data it has to keep track of the regions already visited and to plan efficient paths to unknown regions. The main challenge in designing an efficient algorithm for exploration, therefore, is to find an unified representation that contemplates the diverse aspects of the problem.

Recently, we proposed an extension of the path planner based on harmonic functions developed by Connolly and Grupen[1] for exploration of unknown and unstructured environments [2]. Our approach is based on the use of potential fields computed from relaxation techniques similar to value iteration, applied to a modified version of the grid-based map proposed by Borestein and Koren[3].

As in the case of navigation algorithms [4], exploration algorithms can be classified in three categories according to the emphasis they put in solving one of the following issues: (1) Does it provide a complete (or correct) exploration and mapping of the environment? (2) Is the path followed by the robot optimal in any sense? For instance, is it the minimal possible path? (3) How computationally hard is the algorithm? Can it be successfully implemented in computationally limited agents?

Our work provides a solution that satisfies requirement (1). Furthermore, it generates smooth trajectories that avoid collision with the obstacles while at same time navigating the robot towards the frontier with the minimal navigation path, therefore also providing a fairly good solution for issue (2). Many of these properties emerge from the use of potential fields that are the solution of a boundary value problem where obstacles and goals define the boundaries, as in the harmonic functions method [1].

Concern (3) is acknowledged to be the main problem for such approaches. Harmonic function calculation is known to be computationally costly. In fact, that has been the main difficulty for widespread use of the harmonic technique in path planning tasks, specially in high dimensional c-space. In this paper we demonstrate that, at least for exploration in two-dimensional space, we can reduce enormously the computation cost of the technique without losing the important properties mentioned above. The main idea is to switch between local potential calculations and global potential calculation.

The organization of the paper is as follows. In section 2 we review some of the main features of BVP-exploration. In section 3 we introduce the improvements on the technique that we propose in this paper. In section 4 we explain the implementation. The results are presented in section 5 and discussed in section 6.

II. BVP-EXPLORATION

The BVP-exploration method is an algorithm for map extraction that combines an occupancy grid world representation with a navigation potential calculation on the same grid. The grid structure, therefore, carries both obstacle and path planning information.

The robot is initialized with an empty internal array representing the grid and its attributes. While it travels across the environment it carries along an activation window that indicates the grid cells that are recruited for update, i. e., if a cell is at a given time in the activation window, it becomes part of the explored space and sensor readings are used to update its status. The activation window size can be adaptive and its maximal size depends on the sensor range.

The grid-based representation is suitable for range-finder sensors. In our work we use mostly sonar sensors and we update the certainty of occupancy of a grid cell linearly with the number of observations of an object in that location as in the HIMM method [3]. There are other choices for the update function, for example:Fuzzy [5], Bayesian [6], Gaussian [7], and so on. They are devised to take into account the uncertainty of the sensors readings and they can improve the quality of the generated map. In our method, however, the robot navigates using a potential field that is relatively insensitive to the obstacles fine details, therefore a linear certainty update is enough to guarantee good performances while keeping down computational costs.

The certainty information is thus converted in an occupancy representation through a thresholding operation. The occupancy information is used to build the boundary conditions of the region where the navigation potential field is calculated. The boundary conditions are defined by setting the potential value equal to 1 if a cell is occupied by an obstacle in the HIMM-map sense, i. e., enough observations of an obstacle where attributed to that location; and to 0 if a cell is occupied by a target or it belongs to the frontier of the explored space. For all other cells in the explored region the potential values can be calculated using a suitable discretized partial differential equation [8]. In this paper we use the Laplace's equation [1]

$$\nabla^2 p(\mathbf{r}) = 0 \qquad (1)$$

Navigation is accomplished by descent gradient on the potential surface. The descent gradient points to the direction of the frontier connected by the shortest navigation path. If no obstacle is detected the boundaries of the activated region are zero and the potential is zero in all cells. In this case or in any other situation where there is no gradient to guide the robot, it simply follows the forward direction. Figure 1 illustrates the exploratory behavior generated by our controller. The robot sequentially occupies the positions a, b and c, with the final position indicated by the black circle. Figure 1 also shows snapshots of the internal map at the corresponding positions. At any instant, the vector field points to the closest unexplored place in the environment. In Figure 1 (c) we see how the vector field changes pulling the robot back from a dead-end.

III. Improvements

A. Local-Global Switch

During exploration the potential region grows quite shapeless. In order to update it we define a rectangle

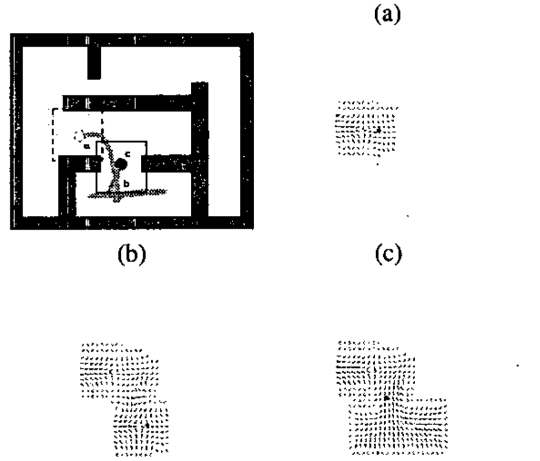


Fig. 1. Exploration Process: On the top the trajectory followed by the robot. (a)-(c) show snapshots of the field and grid configurations for each corresponding position on the trajectory.

window, called **global window** that surrounds the potential region, similar to what is done by Thrun[9]. As the robot explores, the global window increases and so the computational time for the potential calculation, since it is related to its size (global relaxation), see Figure 2.

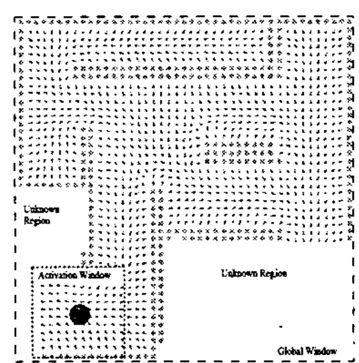


Fig. 2. Adaptive Exploration. The activation window indicate the cell used in the local relaxation. The global window shows the cells used in the global relaxation.

Observe that as an artifact of the array structure the global window can have much more cells than the actual potential region. Diagonal displacements, for example, produce the largest difference between the two. This is not a problem since the spurious non-explored cells introduced in the global window cost only a single "if" operation per iteration. The main computational cost comes from the fact that the number of iterations needed to converge the potential below an error $\epsilon < 10^{-p}$ for a potential region of M cells is at most $\sim p\sqrt{M}$[10].

To minimize some of these effects in [2], [11], we showed that partial relaxation (with a fixed number of iterations) of the potential using Gauss-Seidel method is enough to allow the robot to perform a complete exploration of indoor environment. It works because most of the time the robot is near to an unknown region and explored cells close to this region suffer more boundary influence than farther cells, and consequently converge faster. In these cases, few updates are sufficient to cause the robot be attracted toward the close boundary. Therefore, in practice, the robot is ignoring the situation of cells far from its own position.

This idea can be further improved by noticing that, when the robot is exploring a boundary, or frontier, all that it needs is the relaxation of the potential inside of the activation window, its **local window**, see Figure 2. This window has all the necessary information for exploration: it has known and unknown regions and it has the current robot position.

But exploration using only the local window is effective as long as the robot keeps constantly in contact with the frontiers. This is not the case, for instance, after the exploration of a corridor with a dead-end. Once the dead end is reached the robot has to rely on subtle differences in potential around the border of the activation window to find its way out. That can sometimes results into a temporary oscillatory behavior [11] that resembles a local minimum in the potential. It is not a real local minimum because after a finite number of oscillations, the potential relaxes and the robot is no longer trapped.

In order to avoid this clear decrease in efficiency we propose to switch from local to global relaxation every time the unknown frontier is out of the sensors range, or of the activation window. Since in usual office environments these are relatively rare situations we have an expressive improvement on computational performance, as it is shown later.

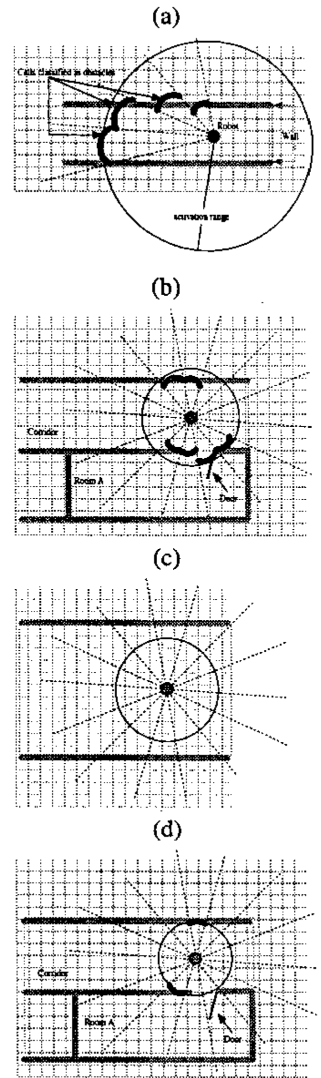


Fig. 3. Activation window. (a)-(c) Influence of fixed activation window. In (a) the corridor is detected as a wall. In (b) an opened door is detected as closed. In (c) the walls do not drive the robot along the corridor. In (d) the robot can detect the correct features of the environment using an adaptive activation window.

B. Adaptive activation window

A second issue addressed in this paper concerns the size of the activation window. We propose to make it adaptive and the advantages are twofold. If a constant activation window is too big, it may cause that narrow passages are wrongly classified as occupied (see Figure 3(a), 3(b)).

If it is too small, as in Figure 3(c), corridors and rooms are viewed as sparse environments and become more complex to navigate [8]. Therefore we adjust the size of the activation window according to the sensor's feedback, Figure 3(d). The size of the activation window is set as the smallest sensor measurement among all sensor returns $\{s_i\}$ at each instant, provided this value is not smaller than a minimum value r_{min} and not bigger than a maximum value r_{max} beyond which the sensors are no longer reliable. Considering $P = \{p_i | 0 \leq i \leq N\}$ the ordered set of sensor's measurements $\{s_i\}$, where $p_i < p_j$ only if $i < j$ and N is the number of sensors, we can write

$$r = min(max(p_0, r_{min}), r_{max})$$

The second advantage of the variable local window is an average reduction in computation time since the window has always the minimum size necessary to allow navigation across the environment.

IV. IMPLEMENTATION

A. Mapping

We define a grid $L_x \times L_y$ to represent the environment. Each grid cell represents a real-world square area and stores the following attributes: **potential** ($p(\mathbf{r})$): represents the potential value in the cell centered in position $\mathbf{r} = (i, j)$; **state**: indicates the cell status, which can be: *not explored*, *free space* or *occupied*. The attribute *not explored* indicates that the position was not visited and its potential value is set to 0. The attribute *free space* indicates that the potential can be updated. The attribute *occupied* indicates that the cell is occupied by an object in the real-world and its potential value is set to 1 in the case of an obstacle and 0 in the case of a goal; **certainty** ($c(\mathbf{r})$): represents the certainty value defined in the HIMM for the cell centered in the position $\mathbf{r} = (i, j)$. The larger this value, the greater the certainty that the object is present.

B. Running

When the experiment begins the robot sets its position equal to (0,0), which corresponds to the center of the 2D grid map. All cells have their property state set to *not explored*, certainty set to 0 and potential to 0.

The sensor detects an object at a distance d from the robot. Using the knowledge about its own position and the direction to the obstacle, the robot increases by 3 the certainty value of each cell in a region defined by $[d - \Delta d, d + \Delta d]$ in the limits of the sensors view cone. The other cells inside the view cone are decreased by 1. A cell has its attribute changed from *free space* to *occupied*, when its certainty value is bigger than 2. This process allows an easy treatment of dynamic and static objects.

After map updating, the potential field is updated for the visited cells using the equation 1 under the usual Dirichlet boundary conditions. The update rule using Gauss-Seidel relaxation method is:

$$p_{i,j}^{new} = \frac{1}{4}(p_{i,j-1}^{new} + p_{i-1,j}^{new}) + \frac{1}{4}(p_{i+1,j}^{old} + p_{i,j+1}^{old}) \qquad (2)$$

A single update of the field consists in iterating the rule above for all grid points either inside the local window or inside the global window for a number of times. After that the negative gradient is computed on the cell containing the current position of the robot. Then the robot moves Δs after adjusting its head direction to the angle θ given by

$$\theta = \arctan(p_{i-1,j} - p_{i+1,j}, p_{i,j-1} - p_{i,j+1}) \qquad (3)$$

where $\arctan(x, y)$ is the inverse tangent taken in the interval $[-\pi, \pi]$ and, in the case of the real robot (Nomad 200), the speed is adjusted according to

$$v = v_{max}\left(0.2 + 0.8\,\frac{90 - |\Delta\theta|}{90}\right), \qquad (4)$$

where $v_{max} = 0.25\ m/s$. The angular correction $\Delta\theta$ is the difference between the robot's orientation and the negative potential gradient direction θ for the current position, reduced to the first quadrant. Therefore its maximal absolute value is 90 degrees, and any turn exceeding 90 degrees in absolute value is implemented as a combination of a $\Delta\theta$ turn and the reversion of the wheels translation. The functional form of (4) and its parameter values were chosen heuristically to cause a reasonably smooth motion.

V. Results

A. Global Exploration x Adaptive Exploration

In this section we present simulation and exploration results that demostrate the usefulness of the improvements in the BVP-exploration introduced in the section III-A. Figure 4 shows the robot exploring an environment with several walls and obstacles disposed in a maze. Figure 4(a) shows the trajectory performed by the robot using the knowledge defined by its global window. Figure 4(b) shows the robot exploring the same environment using the knowledge defined by its adaptive window.

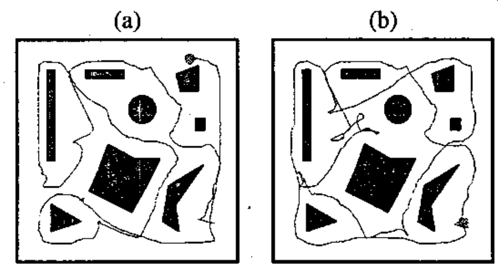


Fig. 4. Global knowledge × adaptive knowledge. a) Exploration guided by global knowledge. b) Exploration guided by adaptive knowledge.

The used strategy does not affect qualitatively the trajectory followed by the robot. In fact, there is a considerable difference in the number of updated cells with time, as showed in Figure 5, for rather equivalent trajectories.

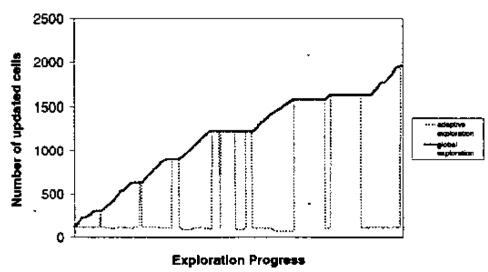


Fig. 5. Number of cells updated during exploration of an indoor environment.

Table I presents the statistics obtained from 10 simulated experiments in a typical environment with different obstacles. The environment has 1m x 1m. Each cell of the map corresponds to 0.02m x 0.02m of the environment

Exploration	$\bar{q}_a$	σ_{q_a}
global	9.2×10^6	2.1×10^6
adaptive	3.6×10^6	1.5×10^6

TABLE I

AVERAGE NUMBER OF CELLS UPDATED DURING THE EXPLORATION PROCESS IN AN ENVIRONMENT (1M X 1M) OVER 10 RUNS.

and the robot is able to sense any object in a range of 10cm. Table I shows the average number ($\bar{q}_a$) of updated cells using adaptive exploration and global exploration and the corresponding standard deviations. The adaptive exploration updates only 39.1% of the number of cell updated under global exploration. Table II shows the average length ($\bar{l}$) of the path followed by the robot using the adaptive and global exploration over 10 runs. In average

Exploration	$\bar{\ell}$	σ_ℓ
global	$15.5m$	$1.4m$
adaptive	$16.3m$	$1.2m$

TABLE II

THE MEAN PATH FOLLOWED BY THE ROBOT DURING THE EXPLORATION OF AN ENVIRONMENT USING GLOBAL AND ADAPTIVE KNOWLEDGE OVER 10 RUNS.

the robot followed a path 4.9% longer using adaptive exploration compared to the global exploration. However, it is important to point out that this increase is modest when compared with the important decrease of 60.9% in the number of cells updated during the exploration.

B. Results on adaptive window

The comparative results between constant and adaptive activation window during the mapping of an environment are presented in this section.

The experiments were done in our lab whose configuration is shown in the Figure 6. The lab has $6.8m \times 6.0m$. Each cell in the map corresponds to $0.1m \times 0.1m$ of the real environment. The maximum speed of robot is $0.15m/s$ and it senses any obstacle in a range of $2m$. The robot is a Nomad200 with a set of 15 sonar sensors and diameter of $0.57m$.

Figure 7(a) shows the robot mapping the environment using an adaptive activation window. Figure 7(b) shows the experiment using a fixed activation window with $2m$ of radius. For this last case, note the quality of the generated map compared to the one obtained with the adaptive window.

VI. DISCUSSIONS AND CONCLUSIONS

The main advantage of our exploration method is that it guarantees that all regions of the environment are thoroughly explored. Changes of the cell status and the

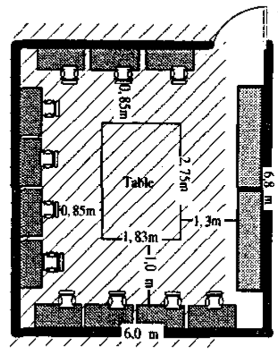


Fig. 6. Laboratory room

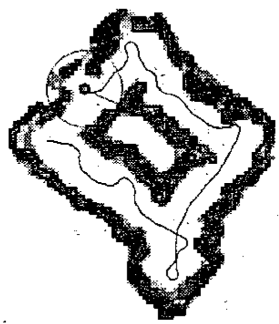
potential guide the robot precisely to missing unexplored spots even in very complex environments. Narrow corridors are very helpful in the process because the repulsion coming from the obstacles keeps the robot centered and the behavior reduces to find the closest unexplored boundary. These boundaries are usually a cross section of a corridor meaning that the robot must proceed exploring the corridor or an opening in the wall indicating that the robot might choose to change trajectory to enter in a new room.

A possible difficulty with the method is the computational cost of keeping the potential updated and relaxed during the progress of exploration. Here we show that if we restrict the global potential relaxation only to the situation where the robot is far from an unexplored frontier it does not reduce significantly the efficiency of exploration while reducing at least 60% in computational time. This strategy can be adapted to other approaches based on numerical potential fields for environment exploration [9], [12] to diminish the cost involved in the exploration process.

Furthermore, we showed that the adaptive activation window is efficient and useful to refine the map automatically. Experiments showed that the adaptive activation window improves the quality of the map because uncertainties of sensor readings are reduced to an acceptable level defined by smallest sensor's return.

A final remark has to be done concerning the problem of localization. Localization is an essential part of map building. In this paper we do not considered this issue. In fact, we rely only on odometry to keep track of the robot's position. But due to the smoothness of trajectory (see Figure 7) obtained from potential field calculation, the odometric error effects were kept very small. Nevertheless we regard the localization problem as an issue of most importance and it will be the next step in the development of our system.

(a)



(b)

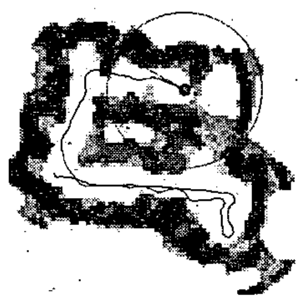


Fig. 7. Experiments using and not an adaptive activation window. (a) With an adaptive activation window. (b) With a fixed activation window with $2m$ of radius.

ACKNOWLEDGMENTS

This work has been partially supported by Brazilian agencies CNPq, FINEP and FAPERGS.

VII. REFERENCES

[1] C.I. Connolly and R.A. Grupen. On the application of harmonic functions to robotics. *J. Robot. and Syst.*, 10:931–946, 1993.

[2] Edson Prestes, Marco A.P. Idiart, Paulo M. Engel, and Marcelo Trevisan. Exploration technique using potential fields calculated from relaxation methods. In *Proc. of IROS2001*, pages 2012–2017, Hawai, EUA, 2001.

[3] J. Borenstein and Y. Koren. Histogramic in-motion mapping for mobile robot obstacle avoidance. *IEEE J. Rob. and Autom.*, 7(4):535–539, 1991.

[4] Nagesware S. V. Rao, Srikumar Kareti, Weimin Shi, and S. Sitharama Iyengar. Robot navigation in unknown terrains: Introductory survey of non-heuristic algorithms. Technical Report ORNL/TM-12410, Oak Ridge National Laboratory, Oak Ridge, Tennessee, july 1993.

[5] G. Oriolo, G. Ulivi, and M. Vandittelli. Fuzzy maps : A new tool for mobile robot perception and planning. *J. Robot. and Systems*, 14(3):179–197, 1997.

[6] A. Howard and L. Kitchen. Generating sonar maps in highly specular environments. In *Proc. 4th Int. Conf. on Control, Automation, Robotics and Vision*, 1996.

[7] A. Elfes. Using occupancy grids for mobile robot perception and navigation. *Computer Magazine*, pages 46–57, June 1989.

[8] Edson Prestes, Marco A.P. Idiart, Paulo M. Engel, and Marcelo Trevisan. Oriented exploration in non-oriented sparse environment. In *Proc. of IROS2002*, pages 2353–2358, Lausanne, Switzerland, 2002.

[9] Sebastian Thrun. Learning maps for indoor mobile robot navigation. *Artificial Intelligence*, 99(1):21–71, 1998.

[10] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes in C: The art of Scientific Computing*. Cambridge University Press, 2nd edition, 1992.

[11] Edson Prestes, Paulo M. Engel, Marcelo Trevisan, and Marco A.P. Idiart. Exploration method using harmonic functions. *Rob. Aut. Systems*, 40(1):25–42, 2002.

[12] E. Morales L. Romero and E. Sucar. A robust exploration and navigation approach for indoor mobile robots merging local and global strategies. In *Proc. of IBERAMIA-SBIA: Adv. in Artificial Intell.*, pages 389–398, 2000.