A Novel Distance Cost Approach for Multi-robot Integrated Exploration

Rafael Gonçalves Colares
Computer Science Department
Federal University of Minas Gerais
Belo Horizonte, Brasil
Email: rcolares@dcc.ufmg.br

Luiz Chaimowicz
Computer Science Department
Federal University of Minas Gerais
Belo Horizonte, Brasil
Email: chaimo@dcc.ufmg.br

***Abstract*—The ability to build reliable maps of unknown environments is important for improving autonomy in robotic systems. Moreover, when multiple robots are able to coordinate themselves to explore different areas of the environment, the exploration efficiency can be greatly improved. In this paper, we propose a novel multi-robot Integrated Exploration technique based on distance cost and frontier exploration. We use a utility function that considers both distance cost and coordination of each robot. Our distance cost technique is based on a function that can be adapted for each environment size and the coordination is integrated with the utility function. Experiments performed with simulated robots show that our approach can significantly reduce the exploration time when compared with other distance cost approaches.**

I. Introduction

The ability to build reliable maps of unknown environments is important for improving autonomy in robotic systems. While exploring unknown environments is a hard task, due to lack of information, SLAM algorithms address this problem with good results. However, if the exploration is performed at random, the efficiency can be reduced. An alternative to that is to use Integrated Exploration techniques to guide the robots during the exploration.

Integrated Exploration techniques add a planning stage to the SLAM algorithm that is responsible to analyze the robot's status and the estimated map and calculate the best action for the robot. These techniques often face a trade-off between visiting already mapped areas in order to improve the map quality and pose estimation or to explore new areas, increasing the covered area to the detriment of the SLAM results.

Since the environment is unknown, the planning stage needs to infer some information about the environment to explore it. One technique uses the distance of the robot to possible goals as a factor to choose the best action. Usually, the distance is treated as the cost to reach a goal, allowing this cost to be different even though the distance is the same, for example, if the robot is at the same position, but facing different angles.

Distance cost is a good choice for Integrated Exploration techniques since it requires only a map, the robots estimated pose and a set of goals, all three available on most SLAM algorithms. Being so, the technique needs only to be able to identify goals and calculate the cost to reach them. The most common distance cost technique is the "frontier exploration" [1]. It consists on finding regions on the map that are on the frontier of explored and unexplored areas to behave as goals. The robot, then, explores the frontier that is closest to its position. For this reason, it is called Near-Frontier Exploration (NFE).

Multi-robot system improves the efficiency of almost every robotic task. When multiple robots explore an unknown environment, they can individually, explore a different area of the environment. But, without any coordination, robots can end up exploring the same area, blocking each other, interfering on sensor readings, etc, resulting on a waste of exploration effort and time. Thus, being able to coordinate the robots is important in order to obtain maximum exploration efficiency.

In this paper, we propose a novel multi-robot Integrated Exploration technique based on distance cost and frontier exploration. We use a utility function that considers both distance cost and coordination of each robot. Our distance cost technique is based on a function that can be adapted for each environment size, allowing a more efficient exploration. We also use a decentralized coordination mechanism, where robots exchange information only when they meet and avoid exploring redundant regions by reducing the utility of frontiers.

This paper is organized as follows. Section II brings a brief literature review on multi-robot exploration. Section III gives an overview of our approach, detailing each part of the system. Experimental results are presented in Section IV, while Section V brings the conclusion and directions for future work.

II. Related Work

The first author to propose an Integrated Exploration technique for multi-robot system was Yamauchi [2]. On its work, Yamauchi [2] control the exploration by using the NFE technique on every robot. Since there is no coordination, robots can explore the same frontier. Aware of this problem, Burgard et al. [3] used a utility function that would reduce the utility of frontier that were assigned for other robots to explore. Although, this seems a simple solution, it requires that the robots share a common map and can communicate at every point of the exploration. On [4], this issue is addressed by keeping a list of the last known position of each robot, and reducing the utility of frontiers based on this list.

Fox et al. [5] proposed a new technique where every robot explores the environment individually, while actively tries to meet with other robots. They use a method to consistently estimate the position of other robots and only exchange information when they meet. When exploring individually, the robots use the NFE to choose the next frontier to explore, but when the encounter happens, the coordination is performed by one of the robots that assigns for each robot on the encounter, a frontier to explore.

The objective of using multiple robots to explore an unknown environment is to spread the robots through the entire environment. To achieve this, Bautin et al. [6] proposes that a robot should only explore a frontier if it is the closest to it. By doing so, if two robots are close to the same frontier, only one will be assigned to it, the other will be assigned to a frontier somewhere else. To reduce the time of calculating distances, a wavefront is started from the frontiers toward the robots. Using this same idea, Kaperski et al. [7] coordinates the robots by segmenting the environment, so that each robot can only explore inside its own segment. The segmentation is done bys starting a wavefront from the robots toward the frontiers. Whenever two wavefronts collide, they stop growing to that direction. Although both techniques successfully spread the robots, they depend on a shared map and a centralized planning. Holz et al. [8] also segment the environment for a single robot exploration, but use a Voronoi diagram to segment the environment.

There are some techniques that use potential fields as a cost metric. These approaches consider frontiers as regions that attract the robots, the closer to a frontier, the higher is the attractive force. Being so, they are NFE-like techniques. Prestes et al. [9] apply the Boundary Value Problem Path Planners (BVP) [10] to create the potential fields. Maffei et al.[11] uses the BVP, but adds a time variant to the potential field.

Other approach to distance cost technique was proposed by Faigl et al. [12]: each robot is assigned to a frontier based on a heuristic that solves the travel salesman problem. They compare their approach using different distance calculation methods for the exploration of the nearest frontier.

It is common on the literature to combine more than one factor to guide the exploration. Juliá et al [13] compares different strategies that combine distance cost with other factors. They compare the strategy from [2] and [3] with [14] that combines distance cost with a utility based on the size of each frontier, Zlot et al. [15] which coordinates the strategy of [14] using market-based algorithms, Makarenko et al. [16] that combines a distance cost together with a utility for the robot localization, and [17] that use behaviors to control the robots. Juliá et al. [13] also state that the best strategy depends on the application.

Our main contribution is to propose a novel distance cost technique based on a utility that leverages some of these approaches in order to improve the quality and efficiency of multi-robot exploration. Robots map the environment individually and only exchange information when they meet each other. For this, a map merging technique is developed. Thus, by combining these ideas, we are able to coordinate a team of robots in a exploration task, where each robot is individually controlled using a utility function based on distance cost and only exchanges information when it meets other robots.

III. Methodology

As mentioned in Section I, our objective is to develop a novel multi-robot exploration system based on distance cost. In our approach, each robot navigates on the environment building its own map, initially without information about other robots. When two robots meet, they exchange information, merge their respective maps together and use this joint information to coordinate.

The exploration is based on a utility function that analyzes the cost to navigate to a frontier. The utility function also coordinates the robots by penalizing frontiers that are close to where other robots were detected. The utility function can be described as:

$$U(f_i) = Dist(f_i) - Coord(f_i) \tag{1}$$

where, f_i is the i^th frontier, U is the utility value, $Dist$ is the distance cost factor and $Coord$ is the coordination factor. The next sections will explain these factors in details.

A. Distance Cost

The idea of exploring the closest frontier, although simple and vastly used, is not always the best solution. By exploring only frontiers close to the robot, little new information about the environment is added to the map. On smaller environments this may be enough, but as the environment size increases, exploring the closest frontier can be a problem. Also, when exploring the closest frontier, the robot tends to never revisit already visited areas, what can cause localization errors.

Taking into account the nature of the application, Distance Cost can have different meanings. For instance, a holonomic robot and a differential robot on the same pose can have different costs to explore the same frontier. In the same way, calculating the cost for a free-obstacle environment can be done using euclidean distance, but for a maze-like environment, this method can be proved to be very ineffective.

To solve these issues, we propose a strategy that uses a function that allows us to choose the best distance cost for each situation. By applying this function to a normalized wavefront [18] starting at the robot's pose towards frontiers, we are able to adapt the distance cost to different environments. This function can be described as:

$$Dist(f_i) = wave(f_i)^{(\alpha-1)} \times (1 - wave(f_i))^{(\beta-1)} \tag{2}$$

where, f_i is the i^th frontier, $Dist$ is the distance cost, $wave$ is the normalized wavefront and the pair (α, β) are variables that allow changing the function form. By changing the value of these two variables, we are able to modify the robots behavior regarding the cost of exploring a given frontier. For instance, if the robot is using pair $(2, 9)$, it will more likely explore a closer frontier than other robot with the pair $(9, 2)$. Figure 1 illustrates the shape of function $Dist$ for different pairs (α, β).

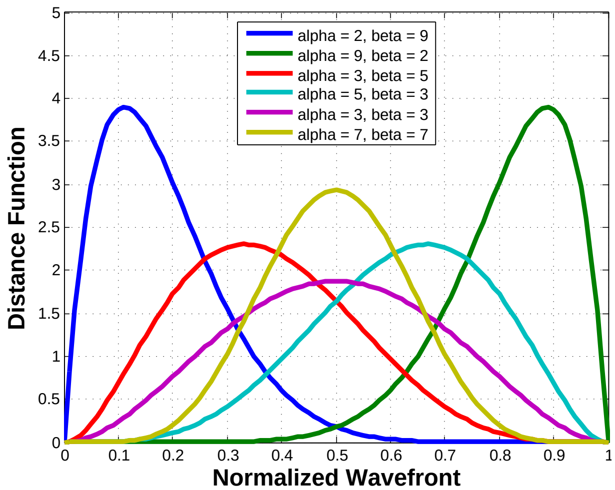

Fig. 1: Distance function for different values of (α, β)

Another effect of the distance cost function can be seem on Figure 1, specially for pairs $(3, 3)$ and $(7, 7)$. Both pairs have their highest value for the frontier that stands on the middle distance from the robot, but they treat the frontiers near this distance differently. If there is no frontier on the middle distance, pair $(3, 3)$ will treat frontiers with normalized distance $\approx (0.3, 0.7)$ with practically the same behavior, while pair $(7, 7)$ is more restrictive to which distance is more important.

B. Coordination

So far, the utility function only takes into account the robot individually, but ignoring other robots can reduce de exploration efficiency. Since we do not have any centralized coordination, the robots must be able to perceive others and coordinate themselves autonomously. To do so, we add a new factor to the utility function that penalizes frontiers near other robots.

When a robot perceives another, this may indicate that the region around the other robot was already explored, and, probably, other regions that are unknown to the robot. Being so, if the robot knows where other robot is, it should merge its own map with the other robot's map and choose to explore a frontier that is far from the other robot. The merging process is explained on Section III-C.

When a robot r_2 is perceived, we calculate its position on robot's r_1 map. At this pose, a wavefront is propagated toward the frontiers. This wavefront differs from the one on Section III-A by the fact that it decreases its value as the wave propagates. For this reason, we call it *negative wavefront*.

The main objective of this negative wavefront is to decrease the utility of a frontier that is close to other robot. This way, on a situation where the best frontier r_1 is likely to explore is next to r_2, we can successfully guide r_1 to explore other frontier, since r_2 will probably explore it.

Figure 2 illustrates the negative wavefront. On Figure 2a there is an estimated map with the pose of another robot detected on it. On Fig. 2b there is the negative wavefront, calculated for that situation, where the brighter the red, the higher the negative value is.

(a) (b)

Fig. 2: (a) The estimated map with the other robot pose and (b) the resulting negative wavefront. Green cells are far from the robot and have lower values, while red cell are close to the robot and have higher values.

C. Map Merge

Our approach does not use the concept of a shared map between robots. When two robots detect each other, it is necessary to merge their estimated maps. To perceive each other, the robots are equipped with image sensors that can detect markers attached on the robots. There are four markers indicating the front, back and sides of the robots. Thus, when a robot spots another, it can infer its relative position and orientation.

When robot r_1 detects robot r_2, they first exchange two pieces of information: their pose, $({}^{r_1}x_{r_1}, {}^{r_1}y_{r_1}, {}^{r_1}\theta_{r_1})$ and $({}^{r_2}x_{r_2}, {}^{r_2}y_{r_2}, {}^{r_2}\theta_{r_2})$, and estimated maps, map_{r_1} and map_{r_2}. Robot r_1 can, also, calculate the pose of r_2, $({}^{r_1}x_{r_2}, {}^{r_1}y_{r_2}, {}^{r_1}\theta_{r_2})$ on map_{r_1}, based on the detected marker.

As r_1 has no information about r_2 starting pose, we need to calculate it in order to transform map_{r_2} to the same frame of map_{r_1}. To do so, we rotate map_{r_2} to map_{r_1} orientation. Basically, we calculate an angle ϕ that aligns both maps into the same orientation. To calculate ϕ we use both robots' orientations (${}^{r_1}\theta_{r_1}$ and ${}^{r2}\theta_{r_2}$) and the relative orientation between robots (${}^{r_1}\theta_{r_2}$).

$$\phi = {}^{r_1}\theta_{r_1} + {}^{r2}\theta_{r_2} + {}^{r_1}\theta_{r_2}. \tag{3}$$

Every robot's starting pose is defined at the center of the map, so, after rotating map_{r_2} by ϕ, we need to translate it, so that the cells on map_{r_2} correspond to the cells on map_{r_1}. To do so, we use the pose where we r_2 was detected

$({}^{r_1}x_{r_2}, {}^{r_1}y_{r_2})$ and r_2 actual pose $({}^{r_2}x_{r_2}, {}^{r_2}y_{r_2})$ to calculate a translation matrix that is applied to map_{r2}.

Now that map_{r_2} is translated and rotated, every cell on map_{r_2} correspond to the same cell on map_{r_1}. Thus, to merge the maps, all that we need to do is to stitch map_{r_2} to map_{r_1}. We do this by keeping all free and occupied cells on map_{r_1} and replacing every unknown cell with free and occupied cells on map_{r_2}. Figure 3 illustrates this process.

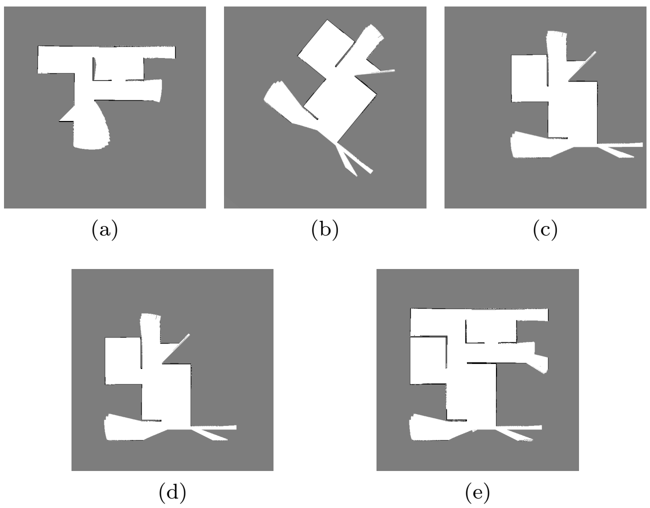

Fig. 3: (a) map_{r_1} and (b) map_{r_2}, respectively. (c) map_{r_2} rotated and on (d) translated. (e) resulting map after the merging process.

IV. Experiments and Results

In order to compare the efficiency of our approach, we compared it with the classical Nearest-Frontier Explorer [2] and MinPOS [6]. The first one only guides the robots to the nearest frontier, without any coordination, but every robot has its own individual map, and only exchanges information when they meet each other. The second approach guides the robots to a frontier, only if it is the closest one to that frontier. This approach uses a shared map and every robot knows each other positions. All three approaches were implemented using the idea of frontier clustering from [8], which clusters frontiers that are close to each other and the robots are assigned to the center of mass of those frontier clusters.

To run experiments we used ROS/Gazebo, simulating two Pioneer-3AT with Sick laser range finder. We used two different environments shown on Fig. 4. On Fig. 4a the environment has $394m^2$ and will be called *small environment* and on Fig. 4b the environment has $812m^2$ and will be called *large environment*.

We compared all three approaches by the time they needed to fully explore the environment. The exploration is considered complete when one of the robots do not have a valid frontier to explore. A valid frontier is one that can be accessed by the robot and that has a minimal size.

In the experiments, we varied the distance cost function to reflect different frontier's distance, by altering the pair

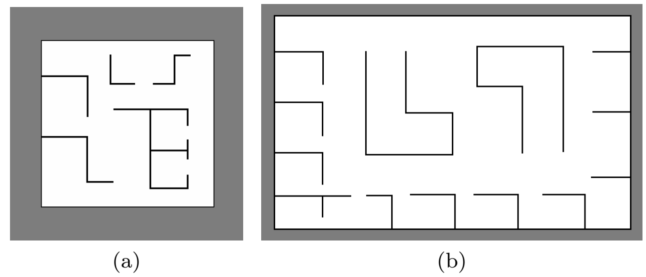

Fig. 4: (a) Small Environment (b) Large Environment.

(α, β). Both values were varied from 1 to 9 in order to prioritize different distances. Every pair was tested 20 times, with random start locations for the robots, and the results presented here are the average of these 20 runs. Since this generates too much information, we also calculated the quality of the estimated map and used it as a ranking mechanism. The map quality metric is based on the work of [13]. First, the estimated map is aligned with a ground truth map of the environment. Then, the maps are subtracted from each other and their differences are counted. This metric indicates the number of cells that have errors on the estimated map, so, the smaller the value, the better is the map quality. The best pairs are the ones with the best (map quality x exploration time) indicator. Since we aim for the lowest exploration time and estimated map error, this indicator is calculated by the distance of the pair to the point $(0, 0)$. Only the best three pairs are shown on the results.

A. Small Environment

In the small environment, the best pairs (α, β) were: $(3, 3)$, $(3, 5)$ and $(5, 9)$. Figure 5 shows where these configurations stand against NFE and MinPOS when comparing the (map quality x exploration time). The figure shows that the map quality variation is practically irrelevant, so the best pairs are the ones with the lowest exploration time. All three pairs are faster than our implementations of NFE and MinPOS.

In order to see the map exploration percentage during the exploration, Figure 6 shows the average of exploration percentage over time. These results show that exploring the nearest frontier for a small environment is not the best solution for reducing the exploration time. It is best to explore the frontiers that are not too close to the robot, but not too far as well.

B. Large Environment

In the large environment, the best pairs were: $(3, 3)$, $(3, 5)$ and $(3, 7)$. Figure 7 illustrates where these pairs stand when comparing (map quality x exploration time) and shows that for the large environment, these pairs had a faster exploration than NFE and MinPOS. Again, the map quality variation was irrelevant when comparing the three pairs with the other two strategies.

Figure 8 shows the map exploration percentage by time. Similarly to the results of the small environment,

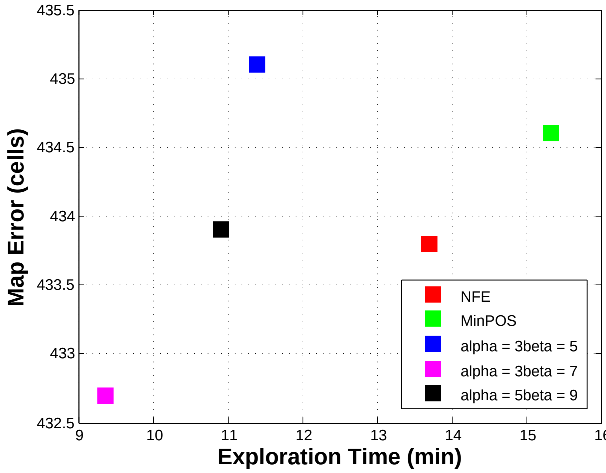

Fig. 5: Exploration Time × Map Quality for the small environment.

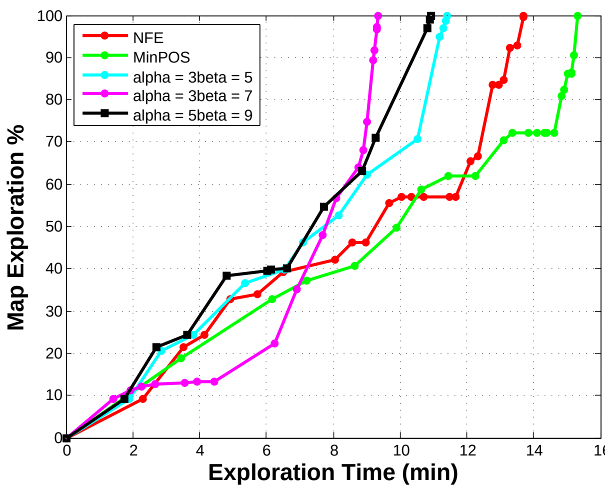

Fig. 6: Map Exploration percentage by the exploration time on the small environment.

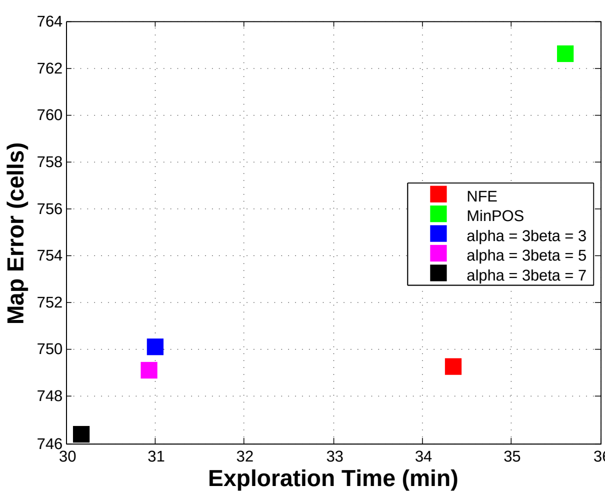

Fig. 7: Exploration Time × Map Quality for the large environment.

the pair $(3, 7)$, which was the faster on both environments, starts with lower exploration percentage than the others and suddenly increases it. This can be explained by the nature of the exploration provided by this pair. It privileges exploring frontiers close to the robot, but not too close and not too far. The idea of exploring the closest frontier emerges form the fact that it is better to make short plans and continuously re-plan, than to plan for long periods of time [16]. By using the pair $(3, 7)$, the robot can make short plans, but still explore a more significant part of the environment than exploring the closest frontier. Finally, when the robots meet, their exchange of information is more significant than the others methods, giving a higher exploration percentage.

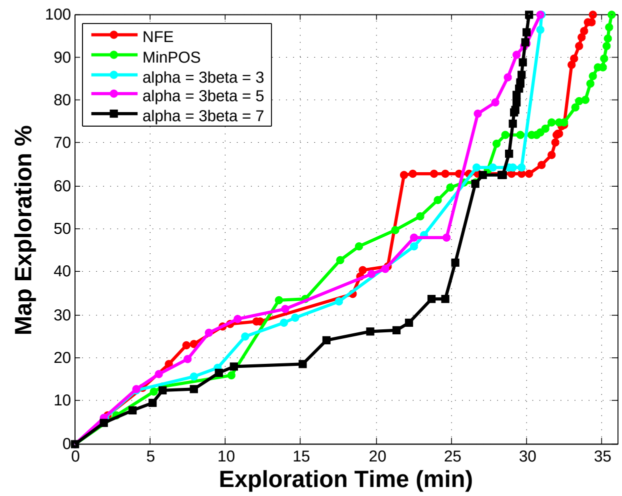

Fig. 8: Map Exploration percentage by the exploration time on the large environment.

To illustrate how the distance cost function behaves for the best pairs, Figure 9 presents the distance cost function for every possible normalized wavefront value. The Figure shows that for pair $(3, 7)$, the robot will choose, preferably, a frontier that is closer to it, but not too close, and not too far. The pair $(3, 5)$ will make the robot choose a frontier that is slightest far from the robot than $(3, 7)$ and, also, this pair is a little more distributed to other distances. Pair $(5, 9)$ is a midterm between $(3, 7)$ and $(3, 5)$, since it prioritizes frontiers distance like the second and really focus on that distance, like the first. Finally, pair $(3, 3)$ was good for the large environment as it aims frontiers that are on middle distance from the robot, but with a higher distribution. This shows that, for large environments, being flexible around the frontiers distance is also important.

V. Conclusion

This paper presented a novel approach for Integrated Exploration with a multi-robot system. By combining both planning and coordination of robots on a single Utility Function, robots are able to explore the environment individually, with low information exchange, and still be able to coordinate themselves. The Utility function combines a function to calculate the distance cost of exploring a frontier and a coordination factor.

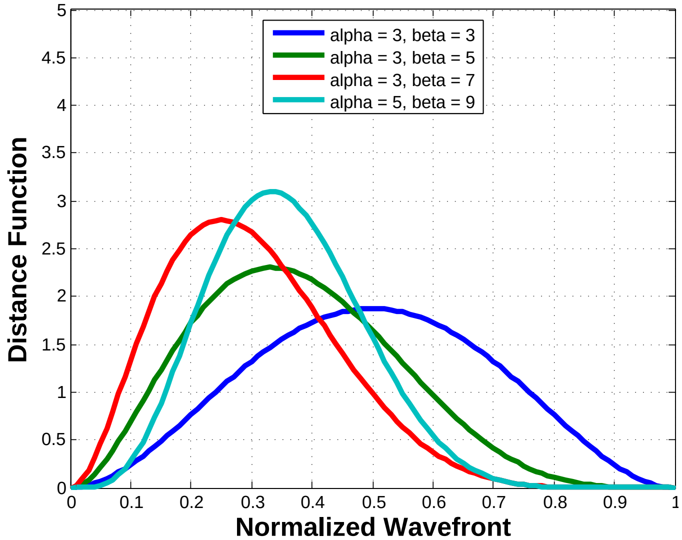

Fig. 9: Distance function for the best pairs (α, β)

We proposed a novel function to calculate the distance cost of frontiers that can adapt for different scenarios. By using the relative distance to the frontier and using a function that can prioritize different distance costs, we were able to efficiently explore the environment. Also, by using a coordination factor directly into the utility function, the robots could explore the environment individually, without the need for a shared map, and still be able to successfully coordinate themselves and merge their maps together.

We compared our approach with the classical NFE and MinPOS. The first one does not have any coordination, only exchanges information when robots meet and explores the closest frontier to the robot. By the results, it is clear that using coordination to the exploration task can result on a faster exploration.

The second one, MinPOS, uses a a centralized coordination system with a shared map for all robots and tries to spread the robots through the environment by making robots explore only frontiers that they are the closest. The results show that using a centralized coordination does not necessarily provides a faster exploration. This is probably due to the time that is needed to the coordination problem. If robots can coordinate themselves individually, this time can be avoided.

Also, the experiments show that planning to explore frontiers that are not too close and not too far from the robot is better than exploring the closest one. This is so, because both are greedy approaches, but the first one can explore a more significant part of the map before replanning the exploration.

As future work, we intend to apply our coordination method to [6]'s approach. This will reduce the coordination overload and we expect to increase the exploration speed. We intend to test our approach with more robots to see how the exploration time behaves.

Acknowledgment

This work was supported by CNPq, Capes and Fapemig

References

[1] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *Computational Intelligence in Robotics and Automation, 1997. CIRA'97., Proceedings., 1997 IEEE International Symposium on*, Jul 1997, pp. 146–151.

[2] ——, "Frontier-based exploration using multiple robots," in *Second Int. Conf. Autonomous Agents*, 1998.

[3] W. Burgard, M. Moors, D. Fox, R. Simmons, and S. Thrun, "Collaborative multi-robot exploration," in *IEEE Int. Conf. Robotics Automation (ICRA)*, 2000, pp. 476–481.

[4] W. Burgard, M. Moors, C. Stachniss, and F. Schneider, "Coordinated multi-robot exploration," *Robotics, IEEE Transactions on*, vol. 21, no. 3, pp. 376–386, 2005.

[5] D. Fox, J. Ko, K. Konolige, B. Limketkai, D. Schulz, and B. Stewart, "Distributed multirobot exploration and mapping," *Proceedings of the IEEE*, vol. 94, no. 7, pp. 1325–1339, 2006.

[6] A. Bautin, O. Simonin, and F. Charpillet, "Minpos : A novel frontier allocation algorithm for multi-robot exploration," in *Intelligent Robotics and Applications*, ser. Lecture Notes in Computer Science, 2012, vol. 7507, pp. 496–508.

[7] S. Kasperski, M. Eich, and J. Wollenberg, "Evaluation of cooperative exploration strategies using full system simulation," in *ICAR '13*, 2013.

[8] D. Holz, N. Basilico, F. Amigoni, and S. Behnke, "Evaluating the efficiency of frontier-based exploration strategies." in *ISR/ROBOTIK*, 2010, pp. 1–8.

[9] E. Prestes and P. Engel, "Exploration driven by local potential distortions," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, 2011, pp. 1122–1127.

[10] E. Prestes and M. Idiartz, "Computing navigational routes in inhomogeneous environments using bvp path planner," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, 2010, pp. 1427–1432.

[11] R. Maffei, V. Jorge, E. Prestes, and M. Kolberg, "Integrated exploration using time-based potential rails," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, 2014, pp. 3694–3699.

[12] J. Faigl, M. Kulich, and L. Preucil, "Goal assignment using distance cost in multi-robot exploration," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, 2012, pp. 3741–3746.

[13] M. Juliá, A. Gil, and O. Reinoso, "A comparison of path planning strategies for autonomous exploration and mapping of unknown environments," *Autonomous Robots*, vol. 33, no. 4, pp. 427–444, 2012.

[14] H. H. Gonzalez-Banos and J.-C. Latombe, "Navigation strategies for exploring indoor environments," *The International Journal of Robotics Research*, vol. 21, no. 10-11, pp. 829–848, 2002.

[15] R. Zlot, A. (tony Stentz, M. B. Dias, and S. Thayer, "Multi-robot exploration controlled by a market economy," in *Proceedings of the IEEE international conference on robotics and automation*, 2002, pp. 3016–3023.

[16] A. A. Makarenko, S. Williams, F. Bourgault, and H. Durrant-Whyte, "An experiment in integrated exploration," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 1, 2002.

[17] M. Juliá, Óscar Reinoso, A. Gil, M. Ballesta, and L. Payá, "A hybrid solution to the multi-robot integrated exploration problem," *Engineering Applications of Artificial Intelligence*, vol. 23, no. 4, pp. 473–486, 2010.

[18] J. Barraquand, B. Langlois, and J.-C. Latombe, "Numerical potential field techniques for robot path planning," in *Advanced Robotics, 1991. 'Robots in Unstructured Environments', 91 ICAR., Fifth International Conference on*, 1991, pp. 1012–1017 vol.2.