Adaptive Robot Formations Using Fast Marching Square Working Under Uncertainty Conditions

Javier V. Gómez
Robotics Lab.
Universidad Carlos III de Madrid
Leganés, 28914, Spain
Email: jvgomez@pa.uc3m.es

Santiago Garrido
Robotics Lab.
Universidad Carlos III de Madrid
Leganés, 28914, Spain
Email: sgarrido@ing.uc3m.es

Luis Moreno
Robotics Lab.
Universidad Carlos III de Madrid
Leganés, 28914, Spain
Email: moreno@ing.uc3m.es

***Abstract*—Robot formations are getting important since they can develop tasks that only one robot could not do or could take too much time. Also, they can perform some tasks better than humans. This paper provides a new algorithm to control robot formations working under uncertainty conditions such as errors in robot positions, errors when sensing obstacles or walls, etc. The proposed approach provides a robust solution based on leader-followers architecture (real or virtual leaders) with a prescribed geometry of the formation and it adapts dynamically to the environment. The algorithm applies the Fast Marching Square (FM^2) method to the path planning of mobile robot formations, which have been proved to work fast and efficently. The FM^2 method is a potential based path planning method with no local minima which provides smooth and safe trajectories. The algorithm described here allows to easily set different behaviours to the formation during its motion depending on the objectives, being possible to set its flexibility. The results presented here show that using this method allows to the formation reacting to either static and dynamic obstacles with an easily changeable behaviour.**

I. Introduction

Due to the multiple applications of robot formations (surveillance, cooperative mapping, etc) the interest on this field has increased enormously during the last years, being actually one of the main topics on robotics research. Right now, a single robot is able to perform very complex tasks on its own but some of these tasks can be perfomed in a more efficent way using a group of robots.

A good enough algorithm to control the motion of a robots formation can influence in very different fields. For example, using a group of small robots allows to carry out a complex task which, if it has to be done with only one robot it could require a very complex design, more expensive and with more limited applications (i. e. transportation of big pieces), thus it is possible to save time and money. When developing exploration or surveillance tasks a robot formation can give better results since the exploration can be done faster or can patrol a wider area, being more effective when developing their objectives.

Another important social impact of using robot formations is in search and rescue objectives. In environmental disasters, such as earthquakes or tsunamis, it is very important to explore all the area as fast as possible, and there are lot of cases where humans are not able to go (i. e. destroyed buildings) so using more than one robot is required. If the robots used are able to keep a formation they can explore more efficently the disaster area finding faster the survivors, creating a map of the destroyed zone (which allows to plan the action to do) or even remove debris without need a very complex robot. Also, in evacuation tasks (i. e. fires) robot formations can be very interesting since they can guide people to the exit using an optimal path, recalculating faster the path when the desired one is impossible. Also, it is important that the robots do not feel, do not doubt and do not fear, allowing them to develop this kind of tasks in a better way than humans.

Therefore applying robot formations to risky or complex tasks can improve their development and can save time, money, increase safety in some kind of jobs and dangerous situations or even save lives.

The algorithm proposed here is focused in leader-follower architecture, where the formation is deformable depending on the position of the robots and also depending on a virtual potential field. The Fast Marching Square (FM^2) method [1] is introduced in robot formations method including new advantages.

The paper is divided in four sections (apart from this one): in Section II the Fast Marching Square algorithm is summarized. How it is applied to robot formations is explained in Section III, where a base algorithm is proposed and some modifications that can be included (avoiding obstacles, changing flexibility, etc.). Impact on society is detailed in Section IV.

July 22, 2011

II. Fast Marching and Fast Marching Square

A. Introduction to Fast Marching and Level Sets

In 1996, J. Sethian proposed the Fast Marching algorithm [2] to approximate the viscosity solution of the Eikonal equation

$$|\nabla(D(\mathbf{x}))| = P(\mathbf{x}) \tag{1}$$

The level set $\{\mathbf{x}/D(\mathbf{x}) = t\}$ of the solution represents the wave front advancing with a media velocity $P(\mathbf{x})$, which is the inverse of the media refraction index $R(\mathbf{x})$. Therefore, the Eikonal equation can be written as $|\nabla(D(\mathbf{x}))| = 1/R(\mathbf{x})$. The resulting function D is a distance function, and if the media velocity P is constant, it can be seen as the Euclidean distance function to a set of starting points, usually the goal points. If the media is non-homogeneous and the velocity P is not constant, the function D represents the distance function measured with the metrics $P(\mathbf{x})$ or the arrival time of the wave front to point $\mathbf{x}$.

B. Key Characteristics of the Fast Marching Square Method

The key characteristics of the FM^2 method are:

- *Absence of local minima.* The way the artificial potentials are calculated assures that the potential given by the distance function $D(x)$ there are no local minima since it is impossible that the wave arrives to a farther place than a given one when to get to that place it is necessary to go through the given one.
- *Fast response.* The planner has to be fast enough to be used reactively in case of unexpected obstacles. A simple treatment of the sensor information and a low complexity order algorithm is necessary.
- *Smooth and safe trajectories.* The trajectories do not need to be refined and they keep a safe distance to obstacles and walls, adapting the velocity to the environment conditions.
- *Reliable trajectories.* The planner provides safe and reliable trajectories avoiding the problem of coordination between local collision avoidance and global planners.
- *Completeness.* As the method consists in the propagation of a wave, it will find a path from the initial position to the goal position if it exists.

C. The Fast Marching Square Algorithm

The FM^2 algorithm can be summarized in the following steps:

1) *Modeling.* Integrates the sensor measurements obtained from a laser scan into a *a priori* grid based map by updating the corresponding occupied cells with black and white information avoiding complex modeling.
2) *Object enlarging.* The objects detected in the previous step are enlarged by the radius of the mobile robot to ensure that the robot does not collide or accepts passages narrower than its size.
3) *FM 1st step.* Using the map obtained after the enlarging, a wave is propagated using the Fast Marching Method from all the points which represents obstacles and walls. The result of this step is a potential map (represented in grey scale in which black represents walls and obstacles and when the cells are farther from them, they become ligther) which can be interpreted as velocity map (or slowness map).
   This step provides a potential similar to the repulsive electric potential (in 2D) and due this analogy, the laws that govern the transmission of electromagnetic waves and light are used to calculate the robot trajectories.
4) *FM 2nd step.* The Fast Marching Method is applied again on the slowness map where the origin of the wave is the goal point and propagates until it reaches the current position of the robot. A new potential $T(x)$ appears representing the propagation of an electromagnetic wave, where the time is added as the third axis in 2D (or the fourth in 3D) and only has one minimum in the current position of the robot. This potential is used to calculate the shortest trajectories in time using the gradient method. In terms of the analogy with electromagnetic waves, the Fast Marching Method solves the eikonal equation that has by solution the fastest path that in optics represents the principle of Fermat: *'Light travels the path which takes least time'*.

III. Control of Robot Formations with Fast Marching Square

The final objective in robot formations is to find the paths and postures (positions and angles) for each robot of the formation, taking into account the characteristics of the environment, the others robots in the formation and the final objective. Then, the robots should be able to carry out a movement adapting the formation to their needs.

In this paper one type of formation is considered: one leader and two followers with a triangle formation. As will be shown, the method described here can be applied to bigger or more complex formations, even using virtual leaders.

The FM^2 method provides a two-level artificial potential which repels the robot from the walls and obstacles. To control robot formation motion it is necessary to add a repulsive force between robots since only working with the artificial repulsive potential given by FM^2 the robots of the formation could crash into each other. Then, integrating the potential given by FM^2 and the repulsive force between robots, each robot has at each moment one single potential attracting it into the objective but repeling it from obstacles, walls and other robots. The main requirement when integrating the all the potentials is to do it in a way that does not create local minima.

A. Description of the Algorithm of Robot Formations Motion using Fast Marching Square

1) Base Algorithm: The FM^2 uses a two-step potential to find out the path: the first step creates a map which can be interpreted as viscosity or slowness map which is named $\mathbf{W}$, and the second step creates a funnel shaped potential, which means the distance to the goal in the metrics $\mathbf{W}$ and is designed as $\mathbf{D}$.

To plan the path for a robot formation using FM^2 the following algorithm is used:

- The environment map $\mathbf{Wo}$ is read as a binary map (0, black, means obstacles or walls; 1, white, means free space). This map is common for either leaders and followers.
- The FM^2 is applied to the leader in order to get a path to the goal point.
- A loop begins, where each cycle represents a step of the robots movement. This loop consists of:
  1) For each cycle, each follower generates a new first potential where the position of other robots is included with uncertainty (see section III-A2).

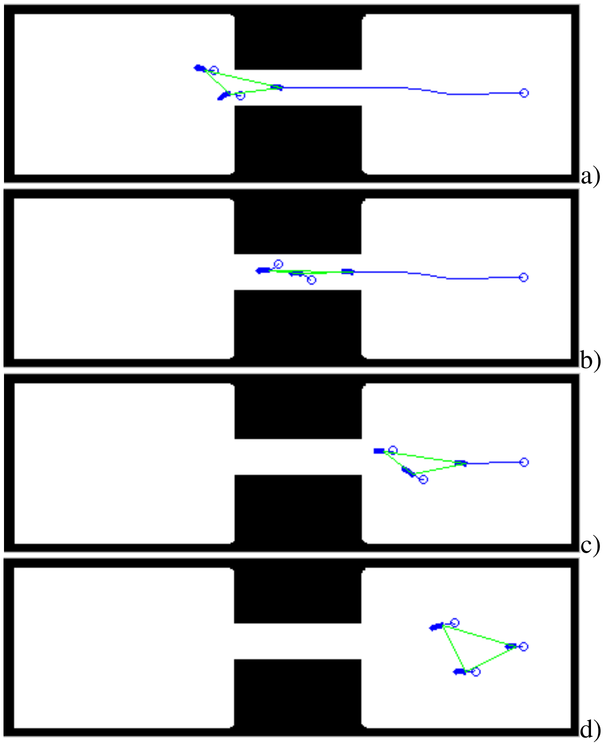


Fig. 1. Sequence of movement of a formation through a narrow corridor.

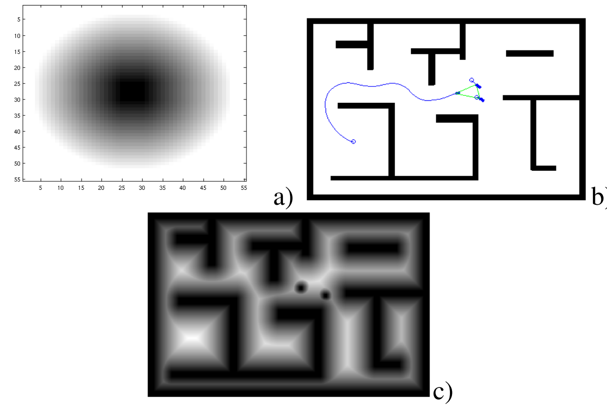


Fig. 2. a) Uncertainty function. b) Actual position of the robot formation. c) First potential of the first follower, where the other two robots of the formation are taken into account using their uncertainty function.

2) From the leader path and the desired formation, a partial goal is calculated for each follower. The distance from the partial goals to the leader path is proportional to the grey level of the partial goals position (lower the grey level is, closer the partial goals are). This is the way that the attractive force between the robots of the formation is implemented. Since each robot treats the others as obstacles they repeal the partial objectives, implementing this way a repulsive force which assures that the robots will never collide.
3) The followers paths are obtained applying the FM^2.
4) All the robots move forward through the obtained path a fixed number of points depending on the refresh rate of the sensors.

The aforementioned algorithm is a base which assures the correct navigation of a robot formation through different environments, avoiding obstacles and adapting the formation in the narrow passages if needed. All the aditional techniques proposed in [3], such as maximum energy configuration, using a tube around the path to decrease the computational cost or adding springs, can be applied to this algorithm with very similar results. Also, this algorithm can be applied to any kind of formations, with real or virtual leaders.

An example of this algorithm is shown in the figure 1.

2) Including uncertainty conditions: Each robot locates itself by means of odometry and localization strategies which are not error-free and have always an uncertainty associated due to sensor noise and measurement errors. The way this uncertainty is dealt is by the use of the fast marching method applied to the other robots of the formations (or *a priori* non-detected obstacles).

This uncertainty is modeled by creating an *uncertainty function* which represent the probability of the robot to be in that point. To create this uncertainty function, the measured robot position is included in a binary map and fast marching is applied, obtaining a grey level map where a darker level means that the probability of find the robot in that point is higher. With this method it is also possible to change the uncertainty level.

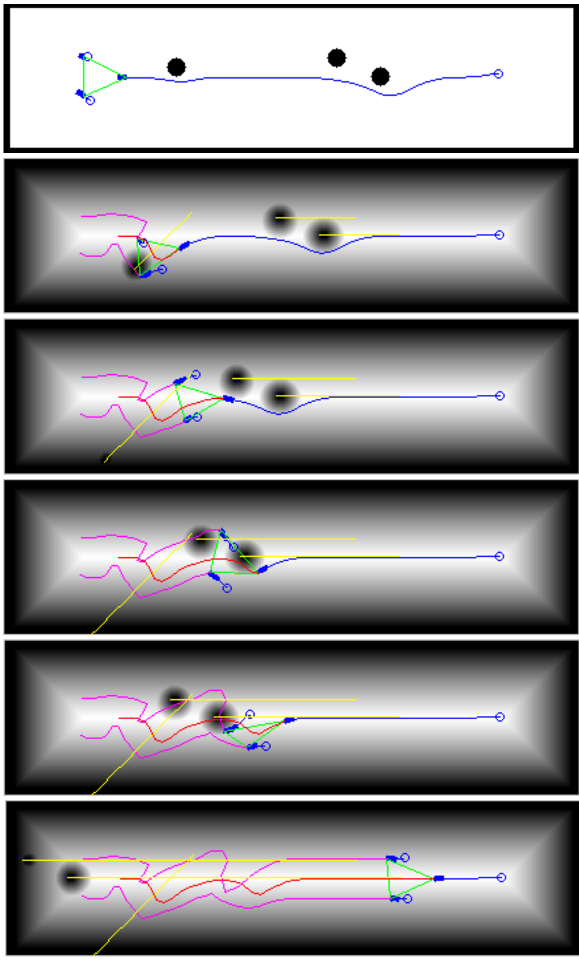
Including an uncertainty function for all members of the formation (leaders or followers), each robot is able to calculate the path to its objective taking into account the global map and the other robots position with its uncertainty included.

This way, the robot will go far from the places were there are not any obstacles but the velocity is slow and it will also go far from the points were the velocity could be high but it is not possible to ensure that in that point there are no other robots or obstacles. The algorithm and this concepts are shown in the figure 2.

3) Mobile obstacles: As well as a single robot has to be able to avoid both static and mobile obstacles, a robot formation has to be able to avoid this obstacles and also adapt the formation to the environment defined by the obstacles at any time. How to overcome this issue depends on the robots, its sensors and on the objective of the formation.

In this case, the mission of the formation is just to be able to reach the destination point as fast as possible keeping the formation and avoiding the obstacles. In this case the leader recalculates its complete path in all iterations so the leader takes into account the obstacles as soon as they are detected. Of course when the leader detects the obstacles, detection errors appears due to sensors noise. This adds uncertainty to the obstacles position, allowing to deal with this problem in the same way as done in section III-A2.

The results of taking into account mobile obstacles with uncertainty are shown in the figure 3.



a)

b)

c)

d)

e)

f)

Fig. 3. a) Binary map with the robot formation and the obstacles. b)-f) Sequence of movement the formation follows to avoid the obstacles.

With the proposed algorithm there is always a difference from the follower positions and their partial objectives. This difference influences on the *flexibility* of the formation since it provides a new degree of freedom which is the distance from the followers to their partial objectives. Setting a short distance keeps the formation rigid, since the paths calculated for the followers have to be shorter but its capability to avoid obstacles decrease. However, setting a long distance from the followers to their partial objectives gives more flexibility to the formation because the FM method explores more environment when calculating the path and better alternatives can be found.

IV. Influence of robot formations on the society

It is possible to say that the proposed algorithm supposes a breakthrough in robot formations since it is robust enough to manage uncertainty conditions and it also provides the safest possible path. Thus, the application of robot formations to real tasks become closer. Some examples of applications and how they influence of society are described:

- *Surveillance.* Moving in a formation, robots can explore more area than only one robot and due their sensors they can do it efficently during the night. Also some surveillance tasks are dangerous and applying robot formations avoids injuries to people.
- *Search and rescue.* As in the previous point, it is possible to explore an area fast with a robots formation, allowing to find survivors faster in disasters and creating a map which can be used by specialists to plan evacuations. Also a robot formation allows to develop the same task with smaller robots than developing the same task with only one robot. This improves the capability of searching in complex areas (i. e. destroyed buildings).
- *Evacuation.* In cases where a massive evacuation is necessary (i. e. fires) a robot formation can help to guide people to the exit and, more over, using the safest possible way thanks to FM^2. A robot formation improves the development of this task since there are more than one robot to follow so smaller groups of people are created improving the efficency of evactuation. Also, the robot formation can be equipped with tools to remove debris, cleaning the path better than using only one robot.
- *Industry and manufacturing.* To manufacture big pieces it could be useful to use a formation of small robots than using only one robot.
- *Bodyguard.* A robot formation can move around a desired point (i. e. human, which acts as a leader of the formation). This improve the bodyguard task since the robots can react faster than humans when the leader is in danger (i. e. an object is moving too fast to the leader or too many people is around him). Also, using robots instead of humans as lifeguards allows to avoid injuries.

All the aforementioned proves that having a good algorithm to control robot formations can improve the development of too many diferent tasks and also allows to improve the safeness of these tasks avoiding injuries and also deaths (as in surveillance or bodyguards) or saving lifes (which is not the same as avoid deaths, as in evacuation tasks). All this also decreases the cost of using human in this tasks: robots can be more expensive than using humans, but robots can be repaired or changed and humans cannot.

References

[1] S. Garrido, L. Moreno, M. Abderrahim, and D. Blanco, "FM2: A Real-time Sensor-based Feedback Controller for Mobile Robots," *International Journal of Robotics and Automation*, vol. 24, no. 1, pp. 3169–3192, 2009.

[2] J. A. Sethian, "A Fast Marching LevelSet Method for Monotonically Advancing Fronts," *Proceedings of the National Academy of Science*, vol. 93, no. 4, pp. 1591–1595, 1996.

[3] S. Garrido, L. Moreno, and P. U. Lima, "Robot Formations Motion Planning using Fast Marching," *Robotics and Autonomous Systems*, vol. In Press, Corrected Proof, pp. 1–25, 2011.