Searching and Tracking for Multi-Robot Observation of Moving Targets

Zheng LIU[1], Marcelo H. ANG Jr.[2], and Winston Khoon Guan SEAH[3]
[1]Dept. of Electrical & Computer Engineering and [2]Dept. of Mechanical Engineering, National University of Singapore, [3]Institute for Infocomm Research, Singapore.
g0202255@nus.edu.sg, mpeangh@nus.edu.sg, winston@i2r.a-star.edu.sg

Abstract: Searching and tracking are important issues in multi-robot observation of multiple moving targets. In this paper, our distributed memory based searching and artificial potential field based tracking algorithms are presented. For searching, a memory table, either local or shared, helps to find targets. For tracking, an artificial potential field based motion control algorithm provides real-time tracking with collision avoidance. Simulations demonstrate the capability of our distributed searching and tracking works.

Introduction

In last two decades, multi-robot cooperative sensing is extensively studied for security, surveillance, and reconnaissance tasks [1]. The mobility of robots advanced traditional static sensor networks, and the cooperation improved the system performance. A typical multi-robot cooperative sensing problem is the "museum" problem: within a large area, given a group of robots whose sensor ability is limited, how to maximize the observation of targets?

Theoretically, centralized control algorithms can be used for museum problem and achieve optimal performance. However, in practice, centralized control is seldom used because the central commander needs to know the information of each robot or node to make decision, also, all possible scenarios have to be predicted and corresponding actions programmed for the robots to react accordingly. Furthermore, centralized control degrades the robustness of the system. When the control canter crashes, the system can not continue to work even though some robots are still functional.

Current research for the museum problem is mostly in distributed manner that enables each robot to control itself based on the locally sensed information or shared information through intercommunications. Parker et al. designed a distributed artificial potential field based control algorithm for Cooperative Multi-robot Observation of Multiple Moving Targets (CMOMMT) [2-3]. Jung et al., [4-5] proposed an optimization algorithm to distribute the robots to enlarge the observable area. In our previous work [6], we presented a conditional weighted artificial potential field based control algorithm, which could achieve robot cooperative searching without a central commander.

In this paper, we present our distributed searching and tracking algorithms for the museum problem. In Section 1, we introduce memory table based searching strategy, and in Section 2, we present conditional weighted artificial potential based motion control for tracking. Then, we discuss the simulation and results in Section 3. Finally, we summarize our contributions and introduce our future work in Section 4.

1. Memory based searching

1.1 The motivation and assumption for using memory

For museum problem, the aim is to maximize the number of the observed targets. Therefore, finding targets is the premise. However, in previous work, such as [2-5], the focus is tracking, and random search is used to find targets. However, to solve museum problem better, the searching part is not trivial and random search is far from satisfactory.

In a real world environment, the movement of visitors is not entirely random. The targets may appear more frequently in some "preferred areas." This is a clue for better searching – search around the preferred areas instead of scattering around randomly and aimlessly. In our approach, we make the following assumptions for our memory based searching algorithm:

- Targets have preferred areas in the environment.
- Robots can locate themselves in the environment. (The technique for localization in indoor environments is already quite advanced now. For example, the AHLoS system [7] can locate an indoor robot with an error in the order of centimeters in magnitude.)
- Robots are controlled in distributed manner. They can share information through intercommunication. However, if intercommunication is not available, they still function.

1.2 The implementation of our memory based searching

1.2.1 Memorization

Because robots can localize themselves in the museum, we partition the entire museum to small grids, and let each robot memorize the target appearance of the grids in its memory matrix F:

- $f_{ij,\,1}$ = target appearance history in grid *ij*
- $f_{ij,\,2}$ = last update time of $f_{ij,\,1}$

Each robot has such a memory matrix, and this matrix is created and maintained by one robot individually. In this matrix, the "target appear history" is a time-related value that increases or decreases depending on the previous observation history of the grid. The matrix *F* is updated in each simulation step. In simulation step *t*, for all grid *ij*, if grid *ij* is observable (within the sensor range of the robot), the corresponding elements $f_{ij,1}$ and $f_{ij,2}$ will be updated by the following rule:

- If target is found: new $f_{ij,1} = +\ preference\_add$ + old $f_{ij,1} * f_{ij,2} / t_{current}$
- If no target is found: new $f_{ij,1} = -\ preference\_sub$ + old $f_{ij,1} * f_{ij,2} / t_{current}$
- Set $f_{ij,2} = t_{current}$

(*preference_add* and *preference_sub* are small positive constants used to update the estimated preference value $f_{ij,1}$.)

Because $f_{ij,2}$ is the last observation update time of grid *ij*, the value of $f_{ij,2}$ will be small compared to $t_{current}$ if the last observation is very old. Therefore, very old observation $f_{ij,1}$ will have little influence when update the $f_{ij,1}$. This is due to the assumption that older information is less reliable.

1.2.2 Intelligent selection

The memory table (matrix *F*) is used to help find the most promising area for searching: for all grid *ij*, calculate the possibilities $p_{ij} = f_{ij,1} * f_{ij,2} / t_{current}$, then find the largest p_{ij}. The corresponding grid *ij* is the most promising area to find a target. This selection is done when the robot needs to search for target. One robot can do this selection by its *F* matrix individually, without the need to communicate with others.

1.3 The sharing of memory among robots

The memory table in Section 1.2 is for a single robot who can only sense a small area around itself. Since the robots are moving and sensing different regions, the *F* matrix is different for each robot because *F* only records the local observation. If all robots can share this matrix, they may have a much larger "view" of the environment. In our approach, we let all robots share matrix *F* to create and maintain a global memory matrix *G* for searching. Obviously, information sharing through intercommunication is the necessary condition for this approach.

The global memory table *G* is generated and updated by collecting the information from the *F* matrices of all the robots. In each simulation step, we update each element g_{ij} of *G* using Eq. 1.:

$$g_{ij} = \frac{\sum_{\text{all robots}} f_{ij,1} * f_{ij,2}}{\sum_{\text{all robots}} f_{ij,2}} \qquad \text{(Eq. 1.)}$$

The denominator is a scaling factor to make the elements in the global memory matrix *G* comparable with those in local memory matrix *F*. It is also used to encourage the robot to explore unobserved area. For instance, early update time will lead to greater *g*, which means unobserved (or observed long ago) area will be more attractive for robots to search.

When a robot needs to search for target, it will calculate out the largest g_{ij} and then move towards the corresponding grid *ij*. In our approach, all robots in search state will go to the same most promising grid. This is based on following considerations:

- Normally, robots are distributed around the museum. If each robot only searches in a promising grid around itself, some areas may not be covered.
- If more than one robot arrive at this most promising grid and they do not find any target there, they will decrease the g_{ij} value much faster than when only one single robot is in this grid. Therefore less time will be wasted in this grid.

2. Artificial potential field based tracking

2.1. Overview of our potential field based motion control

Artificial potential field based control is well known for its simplicity and efficiency because it is reactive and can be implemented in distributed manner. Artificial potential field based motion control assumes that robots (obstacles) and targets carry opposite charges; therefore attractive force exists between robot and target, and repulsive force exists between robots (and obstacles). Based on this assumption, let each robot move under the vector sum of all the forces imposed on it, thus accomplish tracking and avoid collision simultaneously. Obviously, there are two key problems in this control algorithm: (1) Calculating the attractive and repulsive forces (both are called as local forces); (2) Calculating the vector sum of the local forces for a robot.

2.2. The calculation of local forces

In our approach, each robot is assumed to have a panoramic sensor, which can differentiate the sensed objects as obstacles, targets, or robots when the objects are within the circular sensor range around the robot.

We define the orientation of the attractive force is towards the target, and the orientation of the repulsive force is opposite to the obstacle or the other robot.

The magnitude of the local forces is shown in Fig. 1. ($R_{r\text{-}o}$ is the distance between robot and target, $R_{r\text{-}r}$ is the distance between robots or robot and obstacle)

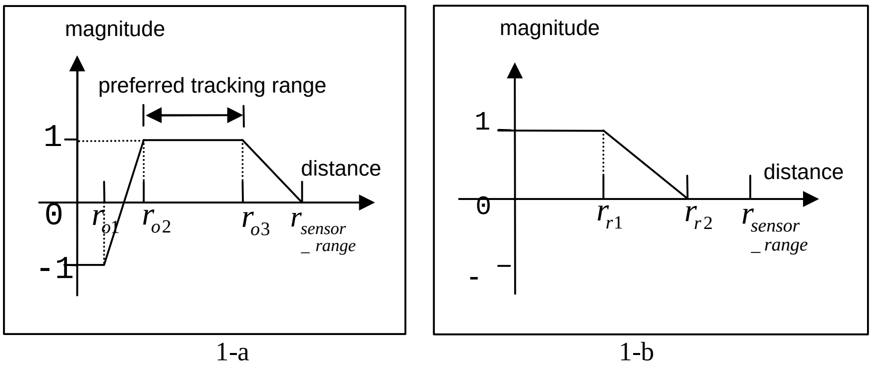


Fig. 1. Magnitude of local forces. (a) attractive force, (b) repulsive force

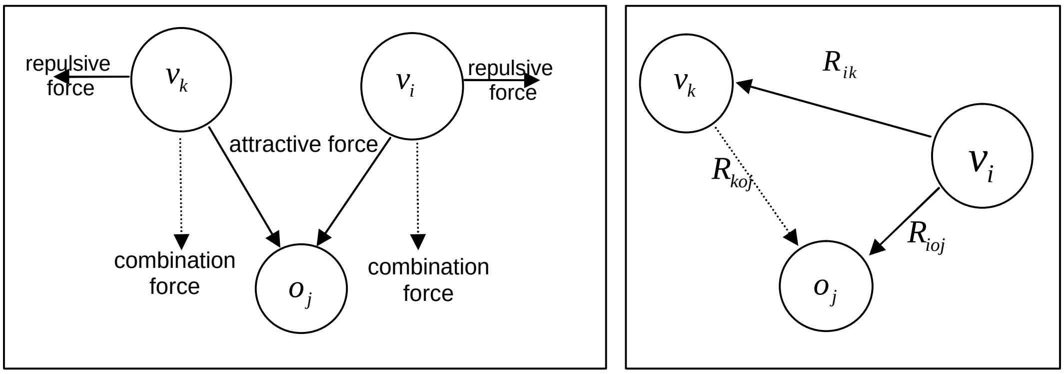
2.3. The calculation of vector sum of local forces

The pure potential field based control equally adds the local forces. However it may degrade the performance of the system. For example, in Fig. 2-a, two robots are following the same target simultaneously. In this case, the two robots and the target will keep a triangle pattern to move until another target or robot appears and disturbs the balance. This is a waste of resources because one of the robots can leave and search for other targets without tracking a target already being followed.

To avoid the deficiency of pure potential field based control, we set weights to attractive forces before adding them up (shown in Eq. 2.). This is the weighted potential field based control algorithm.

$$F_{i\_sum} = \sum_{j=1}^{n} w_{ij} att\_f_{ioj} + \sum_{k=1,k\neq i}^{m} rep\_f_{ik} \qquad \text{(Eq. 2.)}$$

In Eq. 2, n is the number of attractive targets; m is the number of repulsive robots; w_{ij} is the weight of the attractive force between target o_j and robot v_i; att_f_{ioj} is the attractive force to target o_j; rep_f_{ik} is the repulsive force from robot v_k. In Parker's approach [2], whenever another robot was found nearby the target, the weight was decreased. However, in our approach, we only decrease the weight of the attractive force between the farther robot and the target (as shown in Fig. 2-b).



2-a 2-b

Fig. 2. Two robots follow one target simultaneously. (a) triangle pattern, (b) example: decrease weight adaptively: if R_{koj} is smaller than R_{ioj}, weight w_{ij} will be decreased, else w_{ij} unchanged.

Another case needs to be considered is that a robot tracks two oppositely moving targets. In this case, the attractive forces may counteract each other, and then the robot will have little motivation to track. We call this as "hesitation" behavior. When two targets are at the edge of the sensor range, "hesitation" may cause the robot to lose both targets.

To avoid the "hesitation" behavior, we add a restriction to the calculation of the summation of the local forces:

- If at least one target is already within "good" range (a range smaller than the sensor range), the robot will move under the weighted vector sum *Fi_sum* calculated by Eq. 2.
- If no target is within good range, the robot will follow the nearest target, or search for target if no target found.

This restriction will avoid "hesitation" behavior by forcing the robot to give up following one target. We call the control algorithm with such restriction as conditional weighted potential field based control algorithm.

In our simulation for tracking, we will test and compare the performance of these algorithms: (1) Pure potential field based control; (2) Weighted potential field based control; (3) Conditional weighted potential field based control.

3. Simulation and results

3.1. Simulation configuration and methodology

Museum problem is a combination of both searching and tracking. However, the objectives are not the same. For searching, short search time is the aim; whereas for tracking, long tracking time is preferred. Therefore, it is reasonable that we run simulations to test the searching and tracking algorithms separately. The software simulator used is Webots 3, a embedded differential-wheel type robot simulator.

3.2 Searching performance

The simulation configuration for searching is as follows:

- A bounded square area 5.0 x 5.0 m (museum) evenly partitioned to 100 grids.
- 8 targets: move in and out of the museum through 8 doors. On average, each target stays in the museum 1500 simulation steps, and then get out; after 1200 simulation

steps, they reenter. The preferred area is the 9 grids with grid66 as the center.

- 4 robots: they search for the targets, and if a target is found, they track it. However, when a target moves out of the museum, the robot cannot follow it out; it has to search for a new target in the museum again.
- The sensor range of the robot is 0.8m, the preference_add is 0.1; preference_sub is 0.05.
- One simulation episode involves total simulation time of 30000 steps. For one set of parameters, we try 50 episodes to get the average.

The definition of parameters used to evaluate the performance is as follows:

- Prefer_factor (*pf*): *pf* represents the preference level that targets move toward the preferred area. *pf* is calculated out by comparing the real visit frequency and normal visit frequency of the preferred area:

$$pf = \frac{\text{real visit frequency of the preferred area}}{\text{normal visit frequency of the preferred area}}$$

  - Real visit frequency: the real visit frequency during the simulation.
  - Normal visit frequency: the visit frequency if targets have no preference. In our simulation, the normal visit frequency for the 9 grids is the result of 9 times the average visit frequency of all grids.
- Search steps (*ss*): the average steps that a robot needs to find a target

As introduced in Section 1. We compare the performance of our memory based searching in two scenarios: without preferred areas and with preferred areas. We also test the three searching methods: no memory, local memory (*F* matrix), and shared memory (*G* matrix).

For one set of parameters, we run 50 episodes and get the average result shown in Fig. 3.

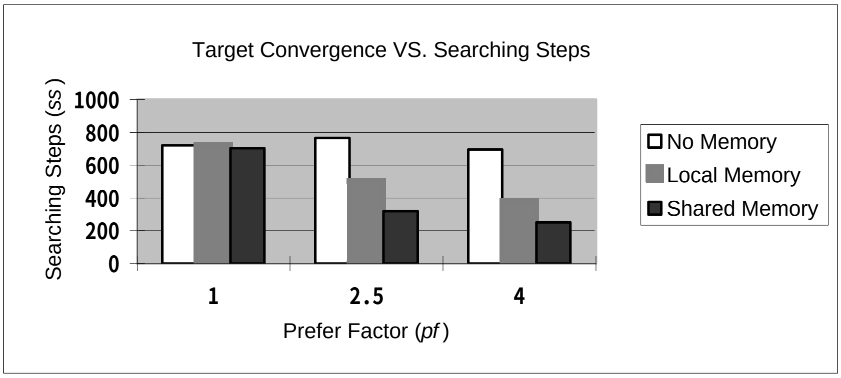


Fig. 3. Searching simulation results

The simulation results show that our memory based searching algorithm works and when the visit preference (convergence) of targets increases, it performs better. However, when targets randomly move in the area, memory becomes useless because there is no rule to predict the appearance of targets. The simulation results also show that the robot team with shared memory is superb to the team without information sharing. A reasonable explanation is that shared memory “enlarges” the observable region of a single robot, and in some sense achieves higher level cooperation among robots.

3.3 Tracking performance

The simulation configuration for tracking is as follows:

- The museum: a 5.0 x 5.0m square area.
- 9 targets move randomly within the museum.
- 3 robots. The sensor radius is 0.5m. Therefore the highest overall sensor coverage is about 9.42% of the entire area.
- For the function defined in Fig. 1-a and 1-b, we select following parameter settings: R_{o1} = 0.1m, R_{o2} = 0.3m, R_{o3} = 0.4m; R_{r1} = 0.2m, R_{r2} = 0.5m.
- Different values of good range are tested: 0.30m, 0.35m, 0.40m, 0.45m, and 0.50m.
- One simulation episode is 45000 steps, and we run 40 episodes for one parameter setting.

We define the system performance *P* as the average number of the targets being tracked, i.e. *P* = 3.2 means in average there are 3.2 targets being tracked during one simulation episode.

In our simulation, we test and compare the system performance under following conditions:

- Pure potential field based control without adaptively setting weight.
- Weighted potential field based control. (the reduction ratio is 0.1)
- Conditional weighted potential field based control.

Fig. 4 and Fig. 5 are the simulation results.

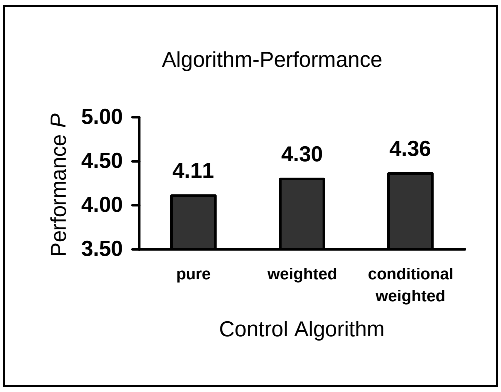


Fig. 4. Performances under different control algorithms

Fig. 5. The relation between the values of good range and the performance

The simulation results show that our potential field based approach is effective. While the total sensor coverage is only 9.42% of the entire area, the system performance *P* of 4.36 is achieved, which means an average of 4.36 targets can be tracked. In addition, the simulation results show that our two modifications to pure potential field based control, adaptively changing weight and setting good range, are effective.

How to find an appropriate size of the “good” range is another problem in our approach. The smaller the value, the smaller the chance that the robot can follow more than one target simultaneously; the bigger the value, the bigger the chance that the robot will present some “hesitation” behavior. From our simulation, we find that about 90% of the sensor range is a nice choice for the value of “good” range.

4. Conclusion

In the paper, we address the problem of searching and tracking of multiple targets using multiple robots. Our distributed searching algorithm relies on the robots remembering the preferred areas for the moving targets. Improved performance is achieved by sharing the memory by intercommunications.

An apparent disadvantage of our memory based searching is that it can hardly handle dynamically changing environments. How to deal with dynamical environments and find a searching strategy is a challenge for our future research.

Once the targets have been searched or are within the sensing range of the robots, the next problem is the tracking of the moving targets. Potential field based control is a simple and scalable real-time algorithm for tracking moving targets and avoiding obstacles. We introduced "weighting" to avoid more than one robot tracking the same target. Furthermore, we have "conditional" weighting to avoid the "hesitation" behavior by the robots. Simulation results show efficacy of the algorithms. However, the local minima problem arises when the attractive and repulsive forces balance and thus the robot cannot move. Some heuristics, such as wall-following method [8], were proposed to solve local minima problem. However, up to now, no general solution exists. Recently, some other algorithms are introduced for the reactive motion control, such as [9], in which the motion of the robot was decided by cooperative and competitive Extended Kohonen Maps (EKMs). Obviously, to design an effective and robust motion control algorithm is one of the future tasks for us.

References

[1] Tucker Balch and Lynne E. Parker, *Robot Teams: From Diversity to Polymorphism*, Natick, Massachusetts: A K Peters Ltd., 2002.

[2] Lynne E. Parker, "Distributed Algorithm for Multi-Robot Observation of Multiple Moving Targets," *Autonomous Robots*, vol. 12(3), May 2002.

[3] Lynne E. Parker and Brad A. Emmons, "Cooperative Multi-robot Observation of Multiple Moving Targets," in *Proc. IEEE Intl. Conf. on Robotics and Automation*, vol. 3, pp. 2082-2089, Albuquerque, New Mexico, USA, 1997.

[4] Boyoon Jung, and Gaurav S. Sukhatme, "Cooperative Tracking Using Mobile Robots and Environment-Embedded, Networked Sensors," in *Proc. IEEE Intl. Symposium on Computational Intelligence in Robotics and Automation*, Banff, Alberta, Canada, Jul. 2001.

[5] Boyoon Jung, and Gaurav S. Sukhatme, "Tracking Targets Using Multiple Robots: The Effect of Environment Occlusion," *Autonomous Robots*, vol. 13, pp. 191-205, 2002.

[6] Zheng Liu, Marcelo H. Ang Jr., and Winston Khoon Guan Seah, "A Potential Field based Approach for Multi-Robot Tracking of Multiple Moving Targets," in *Proc. Intl. Conf. of HNICEM,* March, 2003, Manila, Philippines.

[7] Andreas Savvides, Chih-Chieh Han, and Mani B. Srivastava. "Dynamic fine-grained localization in ad-hoc networks of sensors." In *Proc. The Seventh Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 2001)*, pp. 166-179, Rome, Italy, July 2001.

[8] Xiaoping Yun, and Ko-Cheng Tan, "A Wall-Following Method for Escaping Local Minima in Potential Field Based Motion Planning," In *Proc. ICAR' 97*, Monterey, CA, July, 1997.

[9] Kian Hsiang Low, Wee Kheng Leow, and Marcelo H. Ang Jr, "Action Selection in Continuous State and Action Spaces by Cooperation and Competition of Extended Kohonen Maps," in *Proc. 2nd Intl. Joint Conf. on Autonomous Agents & MultiAgent Systems (AAMAS-03)*, Melbourne, Australia, Jul 14-18, 2003.