an Article from

Journal of Advanced Computational Intelligence

4F Toranomon Sangyo Bldg., 2-29, Toranomon 1-chome, Minatoku, Tokyo 105-0001, Japan
Tel. +813-3508-0051, Fax: +813-3592-0648, E-mail: TAE00762@niftyserve.or.jp

Paper:

A Searching and Tracking Framework for Multi-Robot Observation of Multiple Moving Targets

Zheng Liu[*,***], **Marcelo H. Ang Jr.**[**], **and Winston Khoon Guan Seah**[***]

[*]Dept. of Electrical & Computer Engineering
E-mail: g0202255@nus.edu.sg
[**]Dept. of Mechanical Engineering, National University of Singapore
E-mail: mpeangh@nus.edu.sg
[***]Institute for Infocomm Research, Singapore
E-mail: winston@i2r.a-star.edu.sg
[Received June 25, 2003; accepted September 8, 2003]

The "museum problem" is a typical research topic on multi-robot observation of multiple moving targets. The objective of museum problem is to optimize the distribution of robots, such that the maximal moving targets can be observed. In this paper, we present our memory based searching and artificial potential field based tracking framework for museum problem. For searching, a memory table, either local or shared, can help shorten the searching time for targets. For tracking, our artificial potential field based motion control provides real-time tracking of moving targets with collision avoidance. Qualitative simulations demonstrate the capability of our searching and tracking framework.

Keywords: multi-robot, searching, tracking, potential field, memory

1. Introduction

1.1. Overview of Multi-robot Sensor Networks

In last two decades, multi-robot systems are widely studied for cooperative sensor networks, which are used for security, surveillance, and reconnaissance tasks. The mobility of robots advances traditional static sensor networks, and can adapt to dynamic environments.

For multi-robot sensor networks, the main challenge is to optimize the placement of robots (sensors). In [1,2], Mataric et al. introduced an incremental deployment algorithm to distribute robots. The algorithm let robots move into the area one by one, and each robot chose its location using the information gathered by previous robots. However, the incremental deployment algorithm was slow and not scalable, since it had to maneuver robots one by one. It also could not handle dynamic environments, because it assumed the previous observation was stable. To make the deployment scalable and handle dynamic environments, Mataric et al also introduced an artificial potential field (APF) based algorithm [3], in which the obstacles and robots were mapped as repulsive force sources and therefore robots were dispersed by such forces within the whole area.

Another challenge of multi-robot sensor networks is the cooperation problem. Centralized control can achieve cooperation among robots. However, centralized control is not scalable since all possible scenarios have to be predicted and corresponding actions pre-programmed in order for robots to react accordingly. Furthermore, centralized control usually degrades the robustness of the system, e.g., when the control center fails, the system is unable to continue functioning even though some robots are still operational. Consequently, most current approaches focus on distributed multi-robot systems. Compared with centralized control, distributed control can achieve robustness; however, most recently proposed distributed control algorithms fail in their attempts to be both simple and scalable, especially when the task is quite complex [4] and the robot team is huge. More importantly, there seems to be no formal methodology on distributed control and cooperative behavior. However, artificial potential field based methods show promise and is therefore one of the focuses of this paper.

1.2. Background of Museum Problem

Compared with static environment observation, a more challenging task is to observe moving targets. This is often referred to as the Museum Problem (MP) where a group of robots, which have small sensor coverage, observe targets in a large area. Normally, the museum problem can be described as follows:

- The environment (museum) is a large bounded plain area.
- There are some targets (visitors) moving within the environment (museum).
- Robots (security guards) have local limited-range panoramic sensors, which can scan and differentiate objects within a circular area around the robot. The sum of the sensor coverage of all robots (security guards) is far less than the entire area; hence, robots have to move and track targets.
- Robots (security guards) can move within the entire area. They need to avoid collision with targets and other robots.

- When a target is within the sensor range of a robot, we define the target as being observed. To observe targets, a robot must find a target first. Therefore, searching is the premise of tracking and observation. Normally, random search is used; in this paper, we present our memory based searching method.
- When a robot is observing a target, it has to track the target for continuous observation.
- The objective is to maximize the targets that remain under observation by robots. The more the targets are observed, the higher is the performance.

In general, the challenges of the museum problem lie in finding and tracking targets.

1.3. Related Work for Museum Problem

Museum problem is a hot research topic in recent years. Parker et al. proposed an artificial potential field based approach for cooperative multi-robot observation of multiple moving targets (CMOMMT) [5,6]. In their approach, they not only mapped obstacles and robots as repulsive force sources, but also mapped targets as attractive force sources. Therefore, robots could track targets and avoid collision. Jung et al. also introduced a solution to the museum problem. In their approach [7,8], they considered the global distribution of robots, viz. when robots were too convergent in a small area, some of them would leave this area and move to some area with less robots for better observation.

In our previous work [9], we presented a conditional weighted artificial potential field based control algorithm, which could achieve robot cooperation, e.g., letting robots give up uninteresting targets and leave to track other targets.

In this paper, we present the searching and tracking framework for the museum problem. In Section 2, we introduce our memory table based searching strategy, and in Section 3, we present our conditional weighted artificial potential field based motion control for tracking. Following which, we discuss the simulation and results in Section 4. Finally, we summarize our contributions in Section 5 and introduce our future work in Section 6.

2. Memory Based Searching

2.1. The Motivation and Assumption for Using Memory

The objective of museum problem is to observe as many moving targets as possible. However, if no target is within the sensor range of a robot, then obviously, a robot must find a target first, which means that searching is the premise for observation. However, in previous work, such as [5-8], the focus was only on tracking and random search was used to find targets. To solve the museum problem better, we need to develop strategies to make the searching more efficient.

In a real world environment, e.g. a museum, the movement of visitors is not entirely random. The targets may appear more frequently in some "preferred areas", such as show boxes. This is a clue for better searching - search around the preferred areas instead of scattering around aimlessly. However, how does a robot know where the preferred areas are?

If there exist some preferred areas in the museum, and the robot can memorize the target appearance frequency of different areas in the museum, the robot can locate the preferred areas and search targets using this information. This is the basic assumption of our memory based searching.

In general, our assumptions for robot searching are as follows:

- Targets have preferred areas in the whole environment

It is natural that the target appearance may have some convergent areas, such as the gate, or some hotspots.

- Robots can localize themselves in the museum.

The technique for localization in indoor environments is quite advanced now. Robots can locate themselves by wireless transceivers, such as mobile ad hoc systems [10]. The precision is satisfactory, e.g., the AHLoS system [11] can locate an indoor robot with an error of the order of centimeters in magnitude.

- Robots can memorize their previous observations and have enough computational ability to process this information

This depends on the micro controller of the robots. Fortunately, the speed of current micro controllers can do this job well.

Based on the above assumptions, we design a memory based searching strategy to make the searching more efficient.

2.2. The Steps to Implement Memory Based Searching

Our memory based searching has four steps:

- Initialization
- Memorization
- Intelligent selection
- Search

2.2.1. Initialization

This is the first step for memory based searching. In this step, the entire museum is divided into small grids. We partition the square museum into 100 small squares evenly (10*10). The grids are numbered from 00 (bottom left) to 99 (top right). Then, robots can know which grids they are in by localization.

2.2.2. Memorization

In this step, we record the observed appearance of targets. This is the most important step for our memory based searching. This step is always running even though the robot is tracking a target.

For each robot, we define following matrices:

- Matrix L (10*10):

$$l_{ij} = \begin{cases} 1, & \text{if there is a target targets found within grid } ij \\ 0, & \text{otherwise} \end{cases}$$

It should be noted that if grid ij is beyond the sensor range of the robot, l_{ij} cannot be updated.

- Matrix K (10*10):

$$k_{ij} = \begin{cases} 1, & \text{if grid } ij \text{ is within the sensor range of the robot} \\ 0, & \text{otherwise} \end{cases}$$

- Matrix F (10*10*2):

$f_{ij,1}$ = the "target appear history" in grid ij

$f_{ij,2}$ = last update time of $f_{ij,1}$

The "target appear history" is a time-related value that increases or decreases depending on the history of targets found for that grid.

In each simulation step, the robot observes, puts the observation results in matrix L and K, and then updates matrix F for memorizing the observation results. For example, in simulation step t, for all grid ij from 00 to 99, if k_{ij} is 1, which means grid ij is observable, the corresponding elements $f_{ij,1}$ and $f_{ij,2}$ will be updated. The method to update $f_{ij,1}$ and $f_{ij,2}$ is as the following. If l_{ij} is 1, the $f_{ij,1}$ will increase, else $f_{ij,1}$ will decrease. Also, the $f_{ij,2}$ will be set as current simulation step t. We change the increase and decrease value according to the last update time, such that our system is liable to trust recent observation more:

- For increase: new $f_{ij,1}$ = + *preference_add* + old $f_{ij,1}$ * $f_{ij,2}$ / $t_{current}$
- For decrease: new $f_{ij,1}$ = - *preference_sub* + old $f_{ij,1}$ * $f_{ij,2}$ / $t_{current}$

Because $f_{ij,2}$ is the last observation update time of grid ij, the value of $f_{ij,2}$ will be small when the last observation is long ago. Therefore, the old observation $f_{ij,1}$ will have less influence when update the new $f_{ij,1}$. (*preference_add* and *preference_sub* are positive constants)

Memorization step is running in each simulation step even though the robot is tracking. By continuous memorization, each robot can maintain a memory table (matrix F) of its observation, which can help for searching.

2.2.3. Intelligent Selection

Since we now have a memory table (matrix F), we may use this table to help find the most promising area to search for a target. We need an algorithm to process the data in matrix F. This algorithm must meet following requirements:

- It must reflect the visiting preference
- It should consider the temporal property of "memory": older information is less reliable
- It should be computationally efficient
- It should be flexible enough for information sharing among robots (shared memory)

Based on above requirements, we design an algorithm to find the most promising area. For all grid ij from 00 to 99, we calculate all possibilities $p_{ij} = f_{ij,1} * f_{ij,2}/t_{current}$. By choosing the largest p_{ij}, the corresponding grid ij gives the most promising area to find a target. This selection step is running when the robot needs to search for target and can determine the most promising area based on previous observations.

2.2.4. Search

In the search step, the robot will move towards the most hopeful area calculated out in intelligent selection step. This is a motion control problem which needs to create appropriate actuator commands for moving the robot. In our approach, we use artificial potential field based control for this motion control problem in both searching and tracking. We will introduce this problem with the tracking part in Section 3.

2.3. The Sharing of Memory among Robots

The memory table mentioned in Section 2.2 is for a single robot, which can only sense quite a small area around itself. It is useful to let all robots share their observation histories; therefore each robot may have a much larger "view" with the information coming from others. The sharing of memory table is a realistic assumption because wireless communication can easily transfer information among robots. In our approach, we assume all robots broadcast their memory tables in each simulation step; therefore they can generate a global memory table for searching.

We define a global memory table Gij(10*10). In each simulation step, we update the element gij by following Eq.(1).:

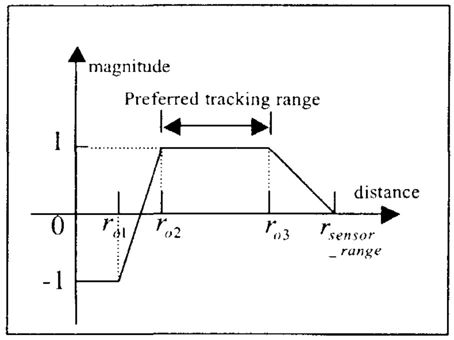
$$g_{ij} = \frac{\sum_{\text{all robots}} f_{ij,1} * f_{ij,2}}{\sum_{\text{all robots}} f_{ij,2}} \quad \cdots\cdots (1)$$

When a robot needs to search for target, it will calculate out the largest gij and then move towards the corresponding grid ij.

In our approach, all robots in search state will go to the most promising grid. This is based on following considerations.

- Normally, robots are distributed around the museum, when they approach this most promising gird, they will follow different paths and therefore scan different parts of the museum. If each robot only search the nearest promising grids around itself, some area 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 gij value much faster than only one single robot is in this grid. Therefore they will go to the next most promising grid quickly, and less time will be wasted.
- In real world, the case that many robots search for targets simultaneously is not very frequently happened. The solution that lets all robots in search state go to one most promising grid is simple but effective in most cases.

1-a



1-b

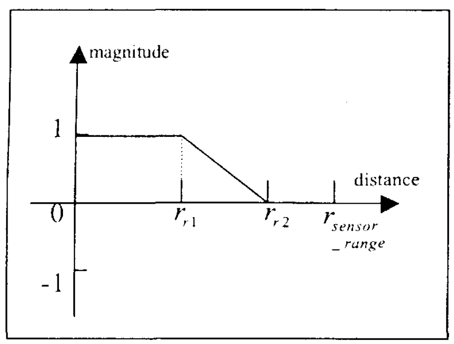


Fig. 1. Magnitude of local forces. (a) attractive force, (b) repulsive force.

2.4. Summary of Memory Based Searching

In this section, we introduce our frame work for searching. The main idea is to let robots remember their observation history and use this information as clue for searching. It should be noted that we assume the targets have preferred areas in the museum. If targets just move around totally randomly, our memory based searching may not be as effective. In simulation and results section, we will compare the performance of our memory based searching in two scenarios: without preferred areas and with preferred areas. We will also test the three searching methods: no memory (random search), local memory, and shared memory.

3. Artificial Potential Field Based Tracking

3.1. Overview of Our Potential Field Based Motion Control

As introduced in the Section 1, artificial potential field based motion control assumes that robots and targets carry opposite charges, therefore there exist attractive force between robot and target, and repulsive force among robots. 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 essential problems in this control strategy:

- Calculating the attractive and repulsive forces (both are called as local forces).
- Calculating the vector sum of the local forces imposed on a robot.

In this section, we will explicitly explain the method we used to solve these two problems.

3.2. The Calculation of Local Forces

In our approach, each robot is assumed to have a panoramic sensor, which can observe and differentiate objects within a circular area around the robot. For example, if target *t1* and robot *r1* are within the sensor range of robot *r2*. Then, robot *r2* can sense their existence, and find the distance and orientation to *t1* and *r1*.

We define that the attractive force is towards the target, and the repulsive force is opposite to the obstacles or other robots.

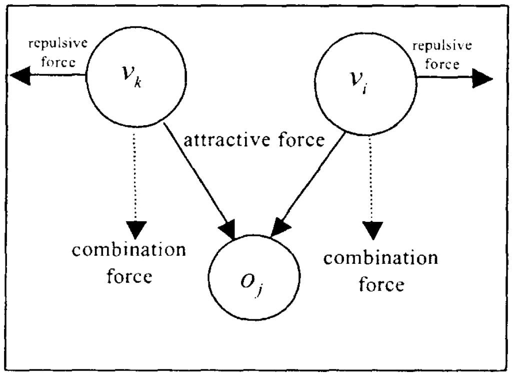
The magnitudes of the local forces are shown in **Fig.1**. **Fig.1a** shows the function defining magnitude of attractive force imposed on a robot due to a target. Let *Rr-o* be the distance between the robot and the target. The magnitude of the attractive force depends on *Rr-o*:

- *Rr-o* <=*Ro1*: The magnitude is set as negative maximum. This is to avoid moving too close to the target, therefore avoiding the collision with the target.
- *Ro1*< *Rr-o* <=*Ro2*: The magnitude changes gradually from negative to positive.
- *Ro2*< *Rr-o* <=*Ro3*: In this segment, the magnitude is set maximum, so that the robot is inclined to keep tracking the target within this range. We call this range as "preferred tracking range".
- *Ro2*< *Rr-o* <=*Rsensor_range*: The magnitude will gradually decrease.
- *Rsensor_range*< *Rr-o*: The magnitude is set as zero because the target is beyond the sensor range of the robot.

Fig.1b shows the magnitude of the repulsive force imposed on a robot due to other robots or obstacles. Let *Rr-r* be the distance between two robots (or robot and obstacle). The magnitude of the repulsive force depends on *Rr-r*:

- *Rr-r* <=*Rr1*: The magnitude is set as maximum. In this segment, the repulsive force is strongest, so that two close robots will move apart rapidly.
- *Rr1*<*Rr-r*<=*Rr2*: The magnitude decreases gradually to zero.
- *Rr2*<=*Rr-r*: In this segment, the magnitude is set zero because two robots are already far apart enough.

2-a



2-b

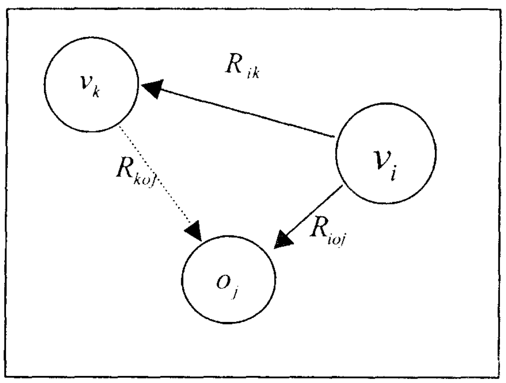


Fig. 2. Two robots follow one target simultaneously. (a) triangle pattern, (b) example: decrease weight adaptively by computing the distance. If *Rkoj* is samaller than *Rioj*, *Vi* will set the weight *Wij*=0.1 or less, else *Wij* = 1.

3.3. The Calculation of Vector Sum of Local Forces

In artificial potential field based motion control, the robot will move under the vector sum of the local forces imposed on it. However, equally adding these forces (pure potential field based control) may degrade the performance of the system. For example, in **Fig.2a**, 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 robot or target appears and disturbs the balance. Obviously, this is a waste of resources because one of the robots can search and follow another target 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. We call this as weighted potential field based control algorithm. In our approach, the summation of

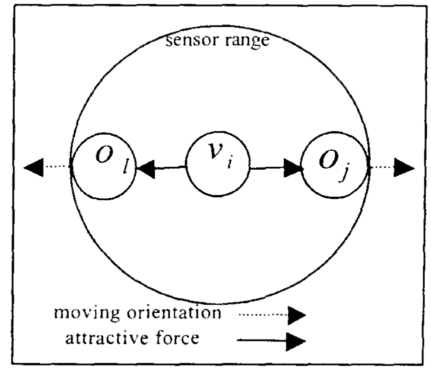


Fig. 3. A robot track two oppositely moving targets.

weighted local forces is as Eq.(2).

$$F_{i\_sum} = \sum_{j=1}^{n} w_{ij} att\_f_{ioj} + \sum_{k=1,\, k\neq i}^{n} rep\_f_{ik} \dots\dots\dots\dots \quad (2)$$

In Eq.(2), m is the number of attractive targets; n is the number of repulsive robots; *Wij* is the weight of the attractive force between target *Oj* and robot *Vi*. Normally, the weight is set as 1 except another robot is also found to follow the same target. In this case, the weight of that target will be decreased to avoid the situation in **Fig.2a**. In Parker's approach [6], the weight was decreased whenever another robot was found nearby the target. But this strategy may lead to a problem: supposing two robots are almost at the same distance from a target, and these two robots find each other. Then, they will both decrease the weight of attractive forces from the target. Finally, maybe both of them will give up tracking the target any more. Obviously, this result is not our expectation.

In our approach, we only decrease the weight of the attractive force between the farther robot and the target. As shown in **Fig.2b**, when robot *Vi* find its peer *Vk* and the target *Oj*, it will compute the distance *Rkoj* between *Vk* and the target *Oj*. Comparing the *Rkoj* and *Rioj*, robot *Vi* will decide whether it needs to decrease the weight of the attractive force to the target *Oj*: If *Rkoj* is larger than *Rioj*, the weight will not change, else the weight will be decreased. The other robot *Vk* will also do the same thing at the same time. Finally, only one robot (*Vi*) will follow the target, and the other one will leave (because of the repulsive force between robots) and search for a new target.

Another case needs to be considered is when a robot is tracking two oppositely moving targets, as shown in **Fig.3**. In this case, the attractive forces will 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. In [6], Parker assumed that the noise of real sensors will avoid this case because the balance of two attractive forces is impossible to happen in real world. However, from our experience, when the moving speed of robot is only a little faster than targets, "hesitation" behavior may badly affect the tracking performance.

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 less 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 move under the strongest attractive force and the repulsive forces, or search for target randomly (when nothing is within the sensor range).

This restriction will avoid "hesitation" behavior by forcing the robot to give up following one target under the condition shown in **Fig.3**. We call this control algorithm with such restriction as "conditional weighted" potential field based control algorithm. In this control strategy, however, we need to decide how to set an appropriate value for the "good" range. If the good range is too small, the performance of the system will be badly degraded because the robot can not follow more than one target before one target is already within the good range. If the good range is too big and near the sensor range, the "hesitation" behavior will still happen.

In our simulation, we will test and compare the performance of these algorithms:

- Pure potential field based control
- Weighted potential field based control
- Conditional weighted potential field based control

4. Simulation and Results

4.1. Simulation Configuration and Methodology

Museum problem is a combination of both searching and tracking. However, their objectives are not the same. For searching, we want to shorten the search time for a target; whereas for tracking, we want to prolong the tracking time of targets. Therefore, the simulation configuration and methodology are different for searching and tracking. In this section, we will present the simulation and results for searching and tracking respectively.

To test our framework, we use Webots, an embedded differential-wheel type robot simulator.

4.2. Searching Performance

The simulation configuration for searching is as follows:

- A bounded square area 5.0 * 5.0 m (museum)
- The area is evenly divided into 100 grids: 10 * 10 (grid 00 to grid 99)
- 8 targets. They move in and out of the area (museum) through 8 different doors around the walls.
- On average, each target will stay in the area (museum) 1500 simulation steps, and then get out of the area; after 1200 simulation steps, they reenter

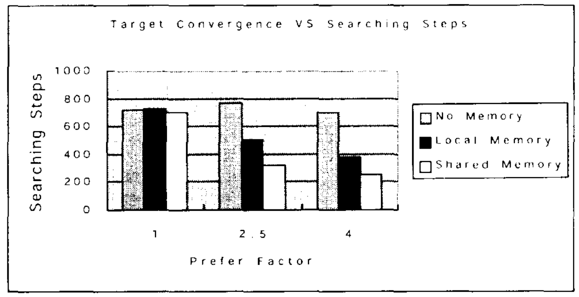


Fig. 4. Searching simulation results.

- When targets are in the museum, they have a preferred area (grid 66), or randomly move.
- 4 robots, always in the area (museum). 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 anymore, it has to search for a new target
- For one mode (no memory, local memory, shared memory), run 20 episodes. Each episode is 30000 simulation step.

The definitions of parameters used for searching are as follows:

- Prefer_factor (*pf*): this parameter represents the inclination that targets move toward the preferred area (grid66). *Pf* is calculated by this formula:

$$pf = \frac{\text{real visit frequency of the 9 grids with grids66 as the center}}{\text{estimated visit frequency of the 9 grids with grid66 as the center}}$$

Real visit frequency: during one simulation step, if a target is within one grid, then the visit frequency of that grid is incremented by one. If two targets are within that grid, then it is incremented by two, and so on.

Estimated visit frequency: evenly distribute the visit frequency into every grid. For example, during simulation, all targets stay in the area for 196062 steps, then the estimated visit frequency of each grid is 196062/100 = 1960.62.

Since grid66 is the preferred area, we made grids 55, 56, 57, 65, 66, 67, 75, 76, 77 as a big preferred area. The real visit frequency and estimated visit frequency are the visit frequency of these nine grids.

Pf represents the preference level of the target toward the preferred area. The larger the *pf*, the more preference that targets want to move to the preferred area (grid 66).

- Search steps (*ss*): the average steps that a robot uses to find a target

When a robot loses its target because the target leaves the museum, it has to search for next target to observe. The times used for finding new targets are summed and the average of this summation is *ss*. The shorter the *ss*, the higher the performance is.

As introduced in Section 2. We want to compare the performance of our memory based searching in two scenarios: without preferred areas and with preferred areas. We also want to test the three searching methods: no

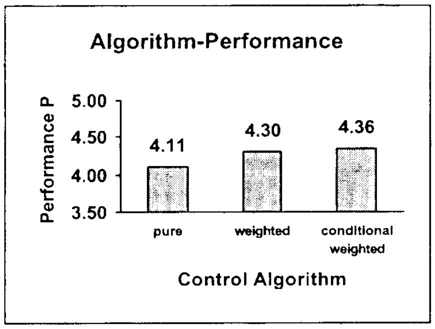


Fig. 5. Performances under different control algorithms.

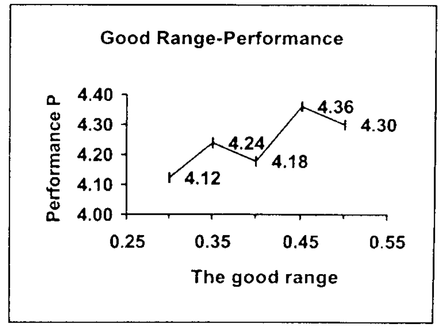


Fig. 6. The relation between the values of good range and the performance.

memory, local memory, and shared memory.

After many simulation runs, we get the results shown in **Fig.4**.

The simulation results shows that when targets have no preference (prefer factor = 1), the memory based searching is not superior to random search. However, when the convergence of the targets increases, our searching strategy works. In addition, using shared memory is better then only using local memory.

4.3. Tracking Performance

The simulation configuration for tracking is the following:

- The museum: a 5m*5m square area. In our simulation, there is no obstacle within this area.
- 9 targets move randomly within the museum. if the path of a target is blocked by objects, such as the wall, other targets, or robots, it will change its moving direction to avoid collision.
- 3 robots act as the security guard to observe targets. Their sensor ranges are set to 0.5m. Therefore the highest overall sensor coverage is about 9.42% of the entire area.
- For the function defined in **Fig.1a** and **1b**, we select following parameter settings:
  Ro1 = 0.1m, *Ro2* = 0.3m, *Ro3*= 0.4m; *Rr1* = 0.2m, *Rr2* = 0.5m.
- Different values of good range are tested: 0.30m, 0.35m, 0.40m, 0.45m, and 0.50m.

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 the simulation time. Regarding each parameter set, we ran the simulation 50 episodes and got the average performance P. However, in the calculation of P, something should be noted: during the initial period of simulation, the robots need some time to find targets. We called the initial time for searching targets as "warm up" period. To eliminate the influence of "warm up" period, we discard the data of the first 5000 simulation steps in each simulation run. Since each episode is 50000 steps, the performance P is the result of the last 45000 simulation steps.

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

- Pure potential field based control without adaptively setting weight: in Eq.(2), let all *Wij* = 1.
- Weighted potential field based control: in Eq.(2), adaptively decreasing *Wij* by the method introduced in Section 3. In our simulation, the reduction ratio is 0.1.
- Conditional weighted potential field based control: adaptively select one or all the attractive forces to follow by the method introduced in Section 3. In our simulation, the "good" range is set as 0.30m, 0.35m, 0.40m, 0.45m, and 0.50m.

Fig.5 and **Fig.6** show the simulation results. In **Fig.5**, we compare the system performance of the three control algorithms. In **Fig.6**, we compare the influence of the values of "good" range in conditional potential field based control algorithm.

The simulation results show that our conditional weighted potential field based motion control works. Also, the good range about 90% of the sensor range is suitable for tracking.

5. Conclusions and Future Work

5.1. Searching

The simulation results show that our memory based searching approach works. When the convergence of targets increases, our approach performs better. This is due to the assumption of our approach: there exist some preferred areas. When targets totally 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 shared memory works better than the local memory. This is due to the fact that shared memory may "enlarge" the view of a single robot, which is a kind of cooperation.

Because in museum problem, searching is the premise for tracking and observation. With the memory based searching, we can accelerate the searching process, and therefore improve the performance for solving museum problem.

5.2. Tracking

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 can be achieved. In addition, the simulation results also prove 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 also an important problem in our approach. The smaller the value, the smaller chance that the robot can follow more than one target simultaneously, thus loses the advantage of potential field based control. The bigger the value, the bigger the chance that the robot will present some "hesitation" behavior, thus the robots may lose the targets. From our simulation, we find that about 90% of the sensor range is a nice choice for the value of "good" range.

5.3. Future Work

An apparent disadvantage of our memory based searching is that it can hardly handle dynamically changing environments. For example, if the preferred areas continuously change, the memory will be unreliable and will be useless. How to deal with dynamical environments and find a searching strategy is a challenge for the museum problem.

Another challenge of museum problem is how to optimize the placement of robots to have better observation of targets. In our current work, we can let robots cooperative in searching and tracking. However, the cooperation is still in low level, e.g., in searching, our approach just let all robots move to the most promising area to look for targets. This may lead the robots omit some hopeful area. Also, let all robots search in a same area may be a waste of resource. We need to improve the algorithm to let robots behave more cooperatively.

Potential field based control is a simple and scalable real-time algorithm. It is also very flexible because it can handle moving targets and moving obstacles. However, in our current work, the environment cannot include concave obstacles. This is because of the deficiency of potential field based control, the local minima problem. Local minima problem arises when the attractive and repulsive forces balance and thus the robot cannot move. This is often happened when the robot is attracted to a target behind a concave obstacle. Some strategies, such as wall-following [12], are brought out to solve local minima problem of the potential field based control. However, up to now, there exists no general solution for it.

Some other approaches for target tracking are also brought out these years. In [13], the motion of the robot is decided by cooperation and competition of extended Kohonen maps, which seems to address the local minima problem of the potential field based control. In our future work, we also need to consider some other motion control strategies.

References:

[1] Andrew Howard, Maja J Mataric, and Gaurav S Sukhatme, "An Incremental Deployment Algorithm for Mobile Robot Teams," In Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS 2002), volume 3, pp. 2849--2854, IEEE, March 2002.
[2] Andrew Howard, and Maja J Mataric, "Cover Me! A Self-Deployment Algorithm for Mobile Sensor Networks," in Proc. the IEEE Intl. Conf. on Robotics and Automation (ICRA2002), 2002.
[3] Andrew Howard, Maja J Mataric, and Gaurav S Sukhatme, "Mobile Sensor Network Deployment using Potential Fields: A Distributed, Scalable Solution to the Area Coverage Problem," In Proceedings of the 6th International Symposium on Distributed Autonomous Robotics Systems, Fukuoka, Japan, June 25-27, 2002.
[4] G. Sen Gupta, C.H. Messom and Sng H.L., "State Transition Based Supervisory Control for a Robot Soccer System," in Proc. of the First IEEE International Workshop on Electronic Design, Test and Applications (DELTA'02), New Zealand, pp. 338-342, 2002.
[5] 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.
[6] Lynne E. Parker, "Distributed Algorithm for Multi-Robot Observation of Multiple Moving Targets," Autonomous Robots, vol. 12(3), May 2002.
[7] 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.
[8] Boyoon Jung, and Gaurav S. Sukhatme, "Tracking Targets Using Multiple Robots: The Effect of Environment Occlusion," Autonomous Robots, vol. 13, pp. 191-205, 2002.
[9] 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.
[10] Dragos Niculescu and B. R. Badrinath, "Ad hoc positioning system (APS)," in Technical Report LCSR-TR-435, Laboratory of Computer Science Research, Rutgers University, 2000.
[11] 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.
[12] 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.
[13] 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.

Name:
Zheng Liu

Affiliation:
M.Eng. student in National University of Singapore

Brief Biographical History:

2001 Received the B.Eng. degree in Department of Automation from Tsinghua University, Beijing, China

Main Works:

- Multi-robot searching and tracking of multiple moving targets

Name:
Winston Khoon Guan Seah

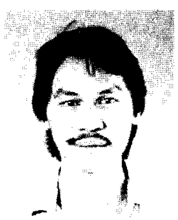
Affiliation:
Lead Scientist and Manager of Networking Department, Institute for Infocomm Research (I2R); concurrently, Adjunct Senior Fellow at the Department of computer Science in National University of Singapore (NUS)

Brief Biographical History:

1987- Received the B.Sc. degree in Computer and Information Sciences from NUS
1993- Received the M. Eng. degree in Electrical Engineering from NUS
1997- Received the Dr.Eng. degree from Kyoto University, Kyoto, Japan
1997- Member of Technical Staff (MTS), Centre for Wireless Communications (CWC), NUS
2000- Senior MTS, CWC, NUS
2001- Principal MTS, CWC, NUS
2002- Principal MTS, Institute for Communications Research, NUS
2003- Lead Scientist and Manager I2R

Main Works:

- Research and development in the areas of mobile ad hoc and sensor networks, mobile/wireless Internet technologies, and Internet quality of service (QoS)



Name:
Marcelo H. Ang, Jr.

Affiliation:
Associate Professor, Department of Mechanical Engineering, the National University of Singapore

Brief Biographical History:

1981- Received the B.S. degrees (Cum Laude) in Mechanical Engineering and Industrial Management Engineering from the De La Salle University, Manila, Phillippines
1985- Received the M.S. degree in Mechanical Engineering from the University of Hawaii at Manoa, Honolulu, Hawaii
1986- Received the M.S. degree in Electrical Engineering from the University of Rochester, Rochester, New York
1988- Ph.D degree in Electrical Engineering from the University of Rochester, Rochester, New York
1989- Assistant Professor of Electrical Engineering at the University of Rochester, New York
1989- Lecturer of Mechanical and Production Engineering at the National University of Singapore

Main Works:

- The areas of robotics, Mechatronics, automation, computer control, and applications of intelligent systems methodologies