Cooperative Multi-robot Target Tracking

Boyoon Jung[1] and Gaurav S. Sukhatme[2]

[1] NavCom Technology, Inc., 20780 Madrona Avenue, Torrance, CA 90503, USA
bjung@navcomtech.com
[2] University of Southern California, Los Angeles, CA 90089, USA
gaurav@robotics.usc.edu

Summary. Target tracking performance can be improved by using multiple robot trackers, but this requires a coordinated motion strategy among the robots. We propose an algorithm based on treating the densities of robots and targets as properties of the environment in which they are embedded. By suitably manipulating these densities a control law for each robot is proposed. The proposed algorithm has been tested through intensive simulations and a real-robot experiment. First, two different versions of the approach were evaluated by studying the performance change as the communication range among robots varies. The results showed that our treatment of the coordination problem is effective and efficient. Second, the developed system was tested on two Segway RMP robots, and the behaviors of the robots in a cooperative tracking experiment provide evidence that the proposed method controls multiple robots appropriately according to the target distribution change.

Key words: mobile robot, multi-robot system, cooperative target tracking.

1 Introduction

Using a mobile robot as a tracking device is beneficial because (1) a mobile robot can cover a wide area over time, which means the number of sensors required for tracking can be kept small, and (2) a mobile robot can re-position itself in response to the movement of the targets for efficient tracking. In cases where the number of targets is much larger than the number of sensors available or when sensors cannot be deployed in advance at the correct locations, mobility is indispensable. Tracking performance can be improved by using multiple robots, and this requires a coordinated motion strategy among robots for *cooperative target tracking*.

The multiple target tracking problem using multiple mobile robots is defined as follows:

Input Estimated poses of M robots and estimated positions of n tracked targets (out of total N targets) in a bounded environment E ($M \ll N$)

Output Motion commands for M robots

Goal Maximize the number of tracked targets n over time T

$$Observation = \sum_{t=0}^{T} \frac{n(t)}{N} \times \frac{1}{T} \times 100 \quad (1)$$

Restriction No prior knowledge of the number of robots or targets, and no target motion model.

The problem seems well-suited to a POMDP optimization framework, and an optimal solution is guaranteed by solving the POMDP problem. However, this approach is not directly applicable to a real world system. The most critical limitation is that the size of the state space ($|E|^{M+N}$) increases exponentially as the number of robots or targets increases. Since the evaluation time of the POMDP problem is exponential in the size of the state space, the problem becomes intractable quickly. Therefore, for scalability, a *distributed* solution is preferable to the centralized, optimal solution. Another limitation is that the optimal policy needs to be re-computed whenever the system configuration changes (examples include adding or removing robots at run time, or adding/removing targets at run time) which implies that the policy computation should be done in real-time. However, most optimization techniques require a significant amount of computation and memory, and they are not suitable to real-time application. Therefore, an *on-line* algorithm is preferable to the off-line, optimal solution.

We propose a *Region-based Approach* as an efficient coordination method, which distributes robots according to the target distribution. In our approach, each robot broadcasts its location and the locations of currently tracked targets. Based on this information and similar information gathered from other robots, each robot independently maintains an estimate of two density distributions - the robot density and the target density. A control law for each robot is generated by using these density estimates. Communication among robots is the key enabler for multi-robot coordination, so the effect of communication range was analyzed; we observed the performance change as the communication range varies. The simulation results show that the proposed algorithm is efficient and robust. The proposed method is also implemented and tested using real robots to validate its applicability to a real-world, resource-constrained system.

The paper is organized as follows. Section 2 summarizes the related work on this topic, and the *Region-based Approach* algorithm is described conceptually in Section 3. Section 4 reports the experimental results and analyzes the performance of the proposed algorithm. The current status and possible improvements are discussed in Section 5.

2 Related Work

Various distributed algorithms have been proposed for multi-robot coordination with applications to multi-target tracking. The ALLIANCE architecture [8] achieves target-assignment by the interaction of motivational behaviors. If a target was not tracked for a while, the robot which was supposed to track the target would give up and another robot in a better position would take up the target. In the BLE architecture [12], if a particular robot thinks it is best suited to track a specified target, it stops other robots from tracking the target by broadcasting inhibition signals over the network. The Murdoch architecture [1] showed that target-assignment problem can be solved using a principled publish/subscribe messaging model; the best capable robot is assigned to each tracking task using a one-round auction.

There have been other approaches that control robot position without explicit target assignment, especially when the ratio of the number of robots to the number of targets is close to 1.0. In [9, 10], the configuration of a team of mobile robots was actively controlled by minimizing the expected error in tracking target positions, and a decentralised system architecture maximizing local information gains was presented in [2]. A reactive motion planner was reported in [7] that maximizes the shortest distance that a target needs to move in order to escape an observer's visibility region.

The Pursuit-Evasion problem introduced in [13] is a formally simplified tracking problem. The goal is to find continuously-moving intruders using a single or multiple searchers with flashlights that emit a single ray of light. [13] presents upper and lower bounds of the number of necessary searchers in a given environment (a simple polygon) and four measures of shape complexity of the environment (the number of edges, the number of reflex vertices, the bushiness, and the size of a minimum guard set). [3] extend the problem to exploit a visible area instead of a single ray of light. Several bounds on the number of pursuers are defined and the complete algorithm for a single pursuer case is presented.

3 Region-based Approach

The *Region-based Approach* is based on the following fundamental assumption:

> *For two comparably sized regions, more robots should be deployed in the one with the higher number of targets.*

Instead of allocating targets to each robot, robots are allocated to each region based on the target distribution and robot distribution. The *robot density* and the *target density* are defined for each position in an environment, and a robot is attracted to (or repulsed from) the position based on those density estimates. For example, the less targets a region has, the less robots the region requires, and the more robots the current region has, the more robots in that region are free to move to other regions. Our approach assumes the following:

Global Localization All robots share a global coordinate system so that the positions of targets detected by different robots can be translated into a single coordinates.

Robust Tracker The cooperative tracking algorithm is decoupled from the low-level target tracker; such a single-robot tracker is described in [5].

Bounded Environment The size of an environment is bounded by the communication range among robots or memory constraints, not by the intrinsic limitation of our algorithm.

3.1 Relative Density Estimates as Attributes of Space

In order to compute robot and target density values at each position, models for robot position, target position, and region boundary are required. Based on the output of the localization algorithm, the position of a robot can be modeled by a delta function or a Gaussian function. When the localization algorithm returns an exact position ($\mathbf{x}_i$) as the best estimate, the robot position is modeled using a delta function (Eqn. 2). When the localization algorithm returns a center position (μ_i) as the best estimate and a covariance matrix (Σ_i) as uncertainty estimate, a bi-variate Gaussian model (Eqn. 3) is adequate. The robot distribution r over an environment is computed by summing the individual models, which are collected through communication among robots at run-time.

$$r = \sum_i \delta(\mathbf{x}_i) \tag{2}$$

$$r = \sum_i N(\mu_i, \Sigma_i) \tag{3}$$

In similar way, the target distribution can be computed. Based on the output format of an underlying target tracker, a delta function model or a bi-variate Gaussian model can be used. The target distribution t over an environment is computed as follows:

$$t = \sum_i \delta(\mathbf{x}_i) \tag{4}$$

$$t = \sum_i N(\mu_i, \Sigma_i) \tag{5}$$

To define the density estimates, a region boundary R of a unit space must be defined. We consider two possibilities: a binary model and a Gaussian model. The binary model (Eqn. 6) defines a region boundary with radius r, which is conceptually simple and computationally cheap. A Gaussian model (Eqn. 7) can be used to define a region boundary when a differentiable output is preferred. The Gaussian distribution is zero-centered and the boundary is determined by a covariance matrix Σ.

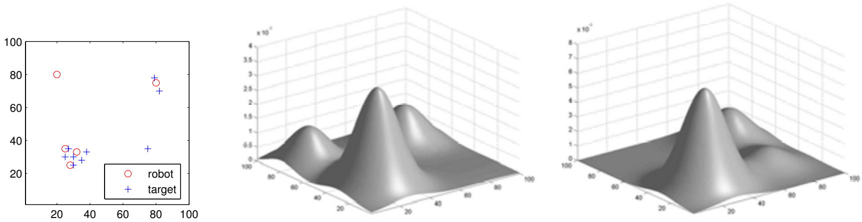
$$R(\mathbf{x}) = \begin{cases} 1.0 & \text{if } |\mathbf{x}| < r \\ 0.0 & \text{otherwise} \end{cases} \tag{6}$$

$$R(\mathbf{x}) = N(\mathbf{0}, \Sigma) \tag{7}$$

The final density distribution of robots (D_r) or targets (D_t) is computed using a convolution of the robot location and region extent:

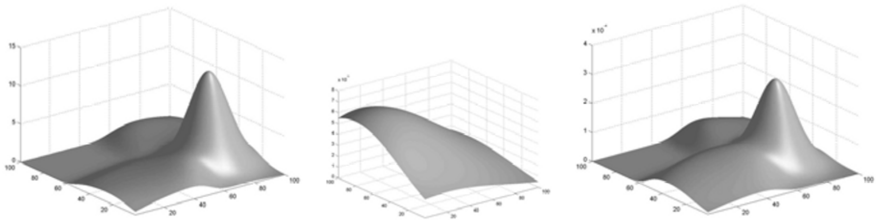
$$D_r(x, y) = r \otimes R = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} r(\tau, \rho) R(x - \tau, y - \rho) d\tau d\rho \tag{8}$$

$$D_t(x, y) = t \otimes R = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} t(\tau, \rho) R(x - \tau, y - \rho) d\tau d\rho \tag{9}$$



(a) object positions (b) robot density distribution (c) target density distribution

Fig. 1. Density distributions



(a) urgency distribution (b) cost function (c) utility distribution

Fig. 2. Utility distribution

Figure 1 (b) and Figure 1 (c) show the final density distribution examples with Gaussian models (Eqn. 3, 5, and 7) when the positions of robots and targets are as shown in Figure 1 (a).

3.2 Urgency Distribution and Utility

Given the distribution of robots D_r and the distribution of targets D_t in a bounded environment, we define the urgency distribution u:

$$u(x, y) = \frac{D_t(x, y)}{D_r(x, y)} \tag{10}$$

Figure 2 (a) shows the urgency distribution calculated using Gaussian models. As shown in Figure 1 (a), there are three groups of targets: six targets around the coordinate $(30, 30)$, two targets around $(80, 80)$, and a single target at $(75, 35)$. The first two groups are being observed by robots, and those regions have relatively low urgency values as shown in Figure 2 (a). However, the last group is not being tracked by any robot, so the urgency value of the region is very high, which means the region 'requires' a robot to migrate towards it.

A cost function c_r for each robot can be combined to compute the final utility function for robot control instead of simply using the urgency distribution as an utility function. For example, the cost of motion can be factored in by multiplying a function which is inverse-proportional to the travel distance. Figure 2 (b) shows an example; the inverse cost function for the robot at $(20, 80)$ has a peak at the current position of the robot since the cost of traverse is zero, and it decreases as it moves further from the current position because the cost of traverse increases. The final utility distribution function is defined as:

$$U(x, y) = u(x, y) \times \frac{1}{c_r(x, y)} \tag{11}$$

It is worth noting that each robot maintains a utility distribution *independently*, and thus each robot would have a different utility distribution from others because of the cost function term c_r. Since the urgency distribution u is calculated using the position information of robots and targets, every robot would maintain the same u distribution when global communication is available. However, the different positions of robots cause different costs for a region, and eventually diverse behaviors for robots are generated.

The final utility distribution for the robot at the coordinate $(20, 80)$ is shown in Figure 2. Intuitively, the region at the coordinate $(75, 35)$ would attract the robot since it has the highest utility value.

3.3 Distributed Motion Strategy

Given the utility distribution, we define two motion strategies. If only local planning is desired, then one possible motor command is a gradient descent method on the utility function as follows:

$$\dot{\mathbf{x}} = -\nabla U \tag{12}$$

If global planning is preferred, then the peak position of the utility distribution can be a goal position:

$$\mathbf{x}' = \arg\max_{\mathbf{x}} U(\mathbf{x}) \tag{13}$$

Each robot plans its motion and executes it *independently* in a distributed manner, and there is no explicit negotiation between robots. However, by sharing the position information of robots and targets, these motion plans are coupled.

4 Experimental Results and Discussion

4.1 Effect of Communication Range

The most critical factor in the performance of the *Region-based Approach* is the communication range among robots. Since each robot estimates the robot and the target densities through communication and the control law is computed based on the estimates, the effectiveness of its motion depends critically on the accuracy of the estimates. Therefore, we studied how the performance of the proposed algorithm degrades as the communication range shrinks through intensive simulations.

The environment was a 50×50 meter sized empty space, and the grid size for the utility function representation was fixed to one meter. The number of robots and targets were fixed to three and twelve respectively, and the target motions were random.

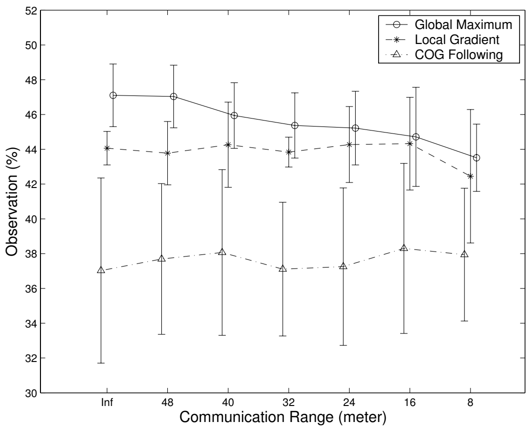


Fig. 3. Performance comparison of three coordination methods

In order to remove the effect of the underlying low-level tracker, an omni-directional, perfect sensor with 8-meter sensing range was assumed. The communication range varied from infinity to 8 meters in steps of the sensor range. Each configuration ran for 10 minutes a total of 10 times, and the average performance was taken as the final result.

Three different coordination methods were compared. The *COG Following* method controls a robot to be positioned at the center of a tracked target group so that the number of tracked targets is maximized locally. There is no communication among robots in this method. The *Local Gradient* method adopts the *Region-based Approach* with the motion strategy in Eqn. 12, which performs hill-climbing on the utility distribution function. Similarly, the *Global Max* method generates a control law based on the Eqn. 13, which controls a robot to move toward the most urgent region constantly.

The experimental results are shown in Figure 3. The *Global Max* method showed the best performance for all configurations; it was clearly shown that its performance degrades in inverse proportion to the communication range as expected. When the communication range was short, there was no significant performance difference between the *Global Max* and *Local Gradient* methods. In contrast, the effect of the communication range on the performance of the *Local Gradient* method was not noticeable except when the communication range was 8 meters. It can be understood that the *Local Gradient* method focuses more on the targets in the vicinity of a robot than those further away. Both methods outperformed the *COG Following* method, which provides evidence that coordination helps, and that the *Region-based Approach* is effective and efficient. It is also notable that the standard deviation of the *COG Following* method is larger than the other methods. This means that it is sensitive to the initial condition and the target motions. The *Region-based Approach* showed more stable performance.

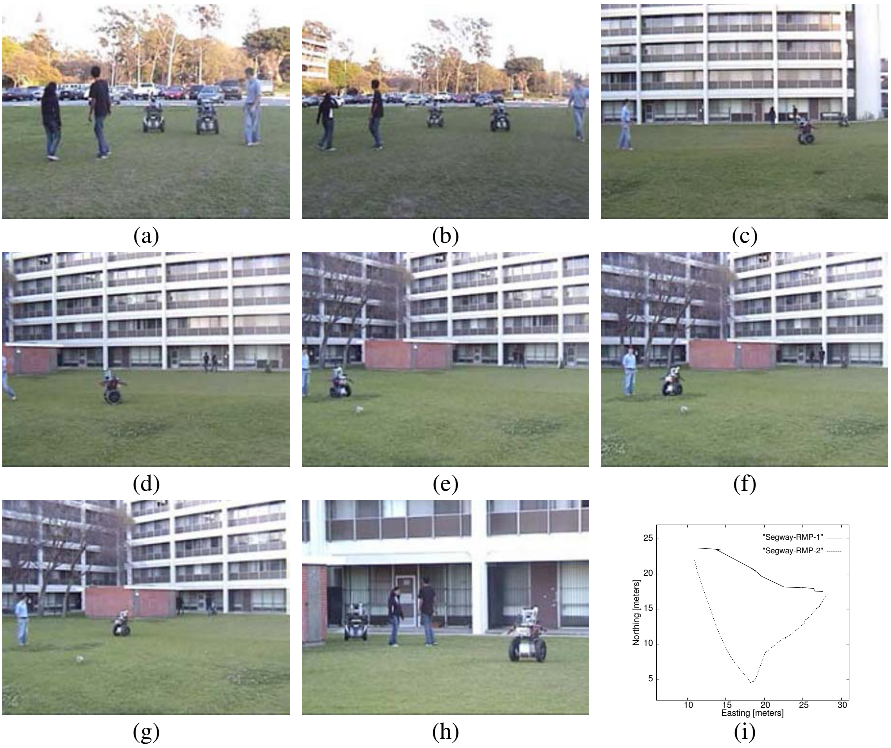


Fig. 4. Snapshots of two Segway RMP robots tracking people cooperatively

4.2 Real-Robot Experiment

The proposed algorithm was implemented and tested using real robots. The system consists of four components: target tracking, localization, cooperative motion planning, and navigation. The target trackers described in [5] were adopted for multiple target tracking. For robot localization, the data from a differential GPS and an IMU were combined using an Extended Kalman Filter. The tracking results (the target positions in a local coordinate system) and the tracker information (the robot pose in the global coordinate system) were broadcast for cooperation over a wireless network. The *Region-based Approach* described in Section 3 was utilized for cooperative motion planning. Due to limited computational power, the delta function models (Eqn. 2 and 4) and the binary model (Eqn. 6) were used to compute the utility distribution. Given its current pose (from the *Localizer* module) and the goal position (selected by the *Region-based Approach*), each robot was programmed to perform point-to-point, safe navigation using VFH+ (Vector Field Histogram +) [11].

The implemented system was tested using two Segway RMP robots. The environment was an open space (30x30 meters). The targets were three pedestrians, and they moved at regular walking speeds in the open area. The grid size of the urgency function representation was fixed to one meter. The behaviors of two robots were inspected while the number of targets changed dynamically.

The snapshots of the two Segway RMP robots tracking people cooperatively are shown in Figure 4. The robots started from the same position, and there were three people walking in front as shown Figure 4 (a). The people split into two groups as shown in Figure 4 (b): two people walking together on the left, and a single person walking in the opposite direction. As a result, two robots also split, and started to track each group respectively. Each robot broadcast its own pose and the position of tracked targets. When the single person stopped moving (Figure 4 (e)), the robot that was tracking the person lost the target and stopped. At this point, the utility value of the robot's position become low, and the robot decided to help the other robot as shown in Figure 4 (f). Finally, the robot arrived in the area whose utility value was the maximum, and helped the other robot track the targets as shown in Figure 4 (h).

The trajectory of the two robots during the experiment is shown in Figure 4 (i). The robots started from the position (28, 17). The first robot tracked the group of two people and moved to the position (11, 24). The second robot tracked the single person and moved to the position (18, 5). When the single person stopped moving, one of the peaks of the utility distribution disappeared, and the second robot moved to the position (10, 22) as a result.

5 Conclusion

In this paper, we proposed an algorithm for multi-robot coordination with applications to multiple target tracking. The proposed algorithm treats the densities of robots and targets as properties of the environment in which they are embedded, and a control law for each robot is generated by suitably manipulating these densities. Since the proposed mechanism is *on-line*, *distributed* and *expandable*, it can be applied for various sensor configurations. For example, a *heterogenous* sensor network can adopt the mechanism with minimal modification, and sensors can be added to (or subtracted from) a tracking network on the fly without stopping operation.

Two experiments has been performed to evaluate the proposed algorithm. First, two different versions of the *Region-based Approach* were compared using various configurations. Since the communication range is the most critical factor for multi-robot coordination, we varied the communication range and investigated the overall performance change. The experimental results showed that both methods outperformed the 'naive' local-following method, and it was clearly shown that our treatment of the coordination problem is effective and efficient. The developed system was also tested on two Segway RMP robots, and the behaviors of the robots in the cooperative tracking scenario provide evidence that the *Region-based Approach* controls multiple robots appropriately according to the target distribution change.

As an implementation issue, the *Region-based Approach* described in this paper can be specialized by exploiting the characteristics of an environment. For example, when the topology of the structured environment is known in advance, the representation of a utility distribution can become discrete and sparse as described in [4].

When the environment is unstructured, the grid-based representation can be adopted as described in [6].

Acknowledgment

This work is supported in part by DARPA grants DABT63-99-1-0015, and 5-39509-A (via UPenn) under the Mobile Autonomous Robot Software (MARS) program, and NSF CAREER grant IIS-0133947.

References

1. Brian Gerkey and Maja J Matarić. Principled communication for dynamic multi-robot task allocation. In D. Rus and S. Singh, editors, *Experimental Robotics*, volume LNCIS 271 of *VII*, pages 353–362, Springer-Verlag Berlin Heidelberg, 2001.
2. Ben Grocholsky, Alexei Makarenko, Tobias Kaupp, and Hugh F. Durrant-Whyte. Scalable control of decentralized sensor platforms. In *International Workshop on Information Processing in Sensor Networks*, pages 96–112, Palo Alto, CA, 2003.
3. Leonidas J. Guibas, Jean-Claude Latombe, Steven M. LaValle, David Lin, and Rajeev Motwani. A visibility-based pursuit-evasion problem. *International Journal of Computational Geometry and Applications*, 9(5):471–494, October 1997.
4. Boyoon Jung and Gaurav S. Sukhatme. Tracking targets using multiple robots: The effect of environment occlusion. *Autonomous Robots*, 13(3):191–205, 2002.
5. Boyoon Jung and Gaurav S. Sukhatme. Detecting moving objects using a single camera on a mobile robot in an outdoor environment. In *International Conference on Intelligent Autonomous Systems*, pages 980–987, The Netherlands, March 2004.
6. Boyoon Jung and Gaurav S. Sukhatme. A generalized region-based approach for multi-target tracking in outdoor environments. In *IEEE International Conference on Robotics and Automation*, pages 2189–2195, New Orleans, LA, April 2004.
7. Rafael Murrieta-Cid, Héctor González-Baños, and Benjamín Tovar. A reactive motion planner to maintain visibility of unpredictable targets. In *the Proceeding of IEEE International Conference on Robotics and Automation*, pages 4242–4247, May 2002.
8. Lynne E. Parker. Cooperative robotics for multi-target observation. *Intelligent Automation and Soft Computing, special issue on Robotics Research at Oak Ridge National Laboratory*, 5(1):5–19, 1999.
9. John Spletzer and Camillo Taylor. Dynamic sensor planning and control for optimally tracking targets. *International Journal of Robotics Research*, 22(1):7–20, January 2003.
10. Ashley Stroupe and Tucker Balch. Value-based observation with robot teams (VBORT) using probabilistic techniques. In *Proceedings of the International Conference on Advanced Robotics*, Coimbra, Portugal, June 2003.
11. Iwan Ulrich and Johann Borenstein. VFH+: Reliable obstacle avoidance for fast mobile robots. In *Proceeding of the IEEE International Conference on Robotics and Automation*, pages 1572–1577, Leuven, Belgium, May 16–21 1998.
12. Barry B. Werger and Maja J. Matarić. Broadcast of local eligibility for multi-target observation. In *Proceedings of Distributed Autonomous Robotic Systems*, pages 347–356, 2000.
13. Masfumi Yamashita, Hideki Umemoto, Ichiro Suzuki, and Tsunehiko Kameda. Searching for mobile intruders in a polygonal region by a group of mobile searchers. In *Symposium on Computational Geometry*, pages 448–450, 1997.