http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 6 (5): 725-732, 2007
ISSN 1812-5638

An Improved Method for Multi-Target Tracking

Zhou Tong, Hong Bing-Rong, Shi Chao-Xia and Piao Song-Hao
Department of Computer Science and Technology,
Harbin Institute of Technology, Harbin, 150001, China

Abstract: We propose a novel distributed algorithm based on local density of robots and targets to multi-target tracking. In this approach each robot computes the numbers and coordinates of neighbor robots within its communicating range and targets within its sensing range based on the latest tracking information. Utilizing these data construct their virtual potential fields. Each robot independently moves to next position according to the sum of force resulted from the fields. The force is multiplied to a weighted factor based on the local density of robots and targets. The performance of the algorithm is evaluated through simulation experiment. Simulation Experimental results indicate that robots are able to distribute themselves appropriately in response to the movement of targets. The algorithm performs better than the artificial potential field approach with the fixed weighted factor for multi-target tracking.

Key words: Multi-target tracking, multi-robot, APF, local density, observation

INTRODUCTION

The first problem is to track the movements of multiple targets navigating in a bounded area of interest in automated surveillance or military reconnaissance systems. A key research issue in these tasks is that of sensor placement to maintain the targets in view. A number of static sensors can be densely placed in advance to ensure adequate sensory coverage of the area of interest. However some factors may prevent static sensory placement in advance. For example, there may be little prior map information of the area to be observed, the area may be sufficiently large that economics prohibit the placement of a large number of sensors, the available sensor range may be limited, or the area may not be physically accessible.

Generally, two methods are adopted for multi-target tracking at present. First, static sensor networks are introduced for multi-target tracking (Spletzer and Taylor, 2003; Li *et al.*, 2002). Second, autonomous sensor-based robots are used for the task. The second method uses control strategy based global information (Boyoon Jung and Sukhatme, 2004, 2006; Li Tzuu-Hseng *et al.*, 2004; Stroupe and Balch, 2003) and local information (Parker, 1999, 2002; Murrieta-Cid *et al.*, 2002). The control strategy based global information costs a great deal of calculation time and lacks real-time character and flexibility. The use of a cooperative team of autonomous robots based local information is a better solution for distributed control in this domain.

Using a group of mobile robots for multi-target tracking is beneficial in the sense that a mobile robot can cover a wide area over time and can reposition themselves in response to targets' movement patterns for more efficient tracking. Especially, when the number of targets is much bigger than the number of sensors available, the mobility of sensors become indispensable. Tracking performance can be improved by using multiple robots and this requires a coordinated motion strategy among robots for cooperative target tracking. The multi-target tracking problem using a group of mobile robots can be treated as a task allocation problem; given a group of agents (robots) and a group of tasks (targets), assign each task to a proper agent so that the overall performance (the total number of tracked targets over time) is maximized.

In this study, our primary focus is on developing the distributed control strategies that allow the team to attempt to minimize the total time in which targets escape observation by some robot team member in the area of interest. We are interested in real-time solutions for the application in unknown and dynamic environments. We propose a variable weighted force vector approach based on artificial potential fields to coordinate robot team members in an uncluttered environment. Utilizing the local density of robots and targets for a weighted factor to adjust the forces exerted on a robot so as to change the velocity vector. This algorithm can make robot team observe more targets. W e program a numeric simulation to compare the performances with other algorithm (no

Corresponding Author: Zhou Tong, Department of Computer Science and Technology, Harbin Institute of Technology, Harbin, 150001, China Tel: 086-0451-86413388

weighted factor of Artificial Potential Fields and static sensor node) in average observation and coverage. The simulation experiment verifies the validity of the algorithm.

PROBLEM DESCRIPTION

The problem of Multi-Target Tracking is defined as follows. Given:
A: a two-dimensional, bounded, enclosed spatial area
R(t): a team of n autonomous mobile robots, R_i, I = 1, 2,...,n, such that robot R_i (t) is located within area A at time t
T(t): a set of m targets, T_l (t), l = 1, 2,..., m, such that target T_l (t) is located within area A at time t
Under the assumptions listed below.

- Every robot has omni-directional observation sensors that are of limited range. The sensing radius is defined as sr.
- The robots have a broadcast communication mechanism that allows them to send (receive) messages to (from) each other within a limited range. The communication radius is defined as cr.
- A robot has self-locating function and can estimate the positions of sensed targets. For example, using GPS and Video camera.
- The robot has maximum velocity $v_{max}(R_i)$. Target's maximum velocity is $v_{max}(T_l)$. $v_{max}(R_i) > v_{max}(T_l)$. This assumption allows robots have an opportunity to collaborate to solve the problem. If the targets could always move faster, then they could always evade the robots and the problem becomes impossible for the robot team.
- The robot team members and target member share a known global coordinate system.

The goal of the robots is to maximize the average number of targets in A that are being observed by at least one robot throughout the mission that is of length T time units. We say that a robot, R_i, is tracking a target when the target is within R_i's sensing range.
Define an m×n matrix B(t), as follows:

$$B(t) = [b_{il}(t)]_{m\times n}$$

$$b_{il}(t) = \begin{cases} 1 & \text{if roboot Ri is tracking target } T_l(t) \text{ in A at time t} \\ 0 & \text{otherwise} \end{cases}$$

Define Observation as an estimated standard for the problem of Multi-target tracking.

$$\text{Observation} = \sum_{t=1}^{t_{max}} \sum_{l=1}^{m} \frac{g(B(t), l)}{t_{max}}$$

Where,

$$g(B(t), l) = \begin{cases} 1 & \text{if there exists an i such that } b_{il}(t)=1 \\ 0 & \text{otherwise} \end{cases}$$

$q_i = (x_i, y_i)$ denotes the coordinate of ith mobile robot, i = 0,1,2…n, $q_i \in R^2$, $s_l = (x_l, y_l)$ the coordinate of lth target, l = 0,1,2…m, $s_l \in R^2$.

The drive force of ith robot is denoted by $u_i \in R^2$, i = 0,1,2…n.

$$\ddot{q}_i = u_i$$

Let q_{ij} denotes the Euclid distance of node I and j, if $q_{ij} = \|q_i - q_j\| \le cr$, the two nodes are neighbor nodes each other.

Coverage is an important estimated standard for robotic sensor networks. Coverage can be calculated by follow formula.

$$\text{Coverage} = \frac{\sum_{t=1}^{t_{max}} \bigcup_{i=1,\dots n} S_i}{S \cdot t_{max}} \qquad (2)$$

Where, S_i is the coverage region area of robot I. S is total area of interest region and n is the number of robots.

APPROACH

The control rule of robot movement: Artificial Potential Field (APF) control is mostly used for Multi-target Tracking in Multi-robot system. The concept of APF is simple: map the targets as sources of attractive force and map the other robots and obstacles as sources of repulsive force (Howard *et al*., 2002; Spears and Gordon, 1999). The repulsive force exerting to a robot is calculated by the distances to other robots within its communication range. The attractive force is derived from the distances to the targets within its sensing range. Then, let the robot move under the vector sum of the attractive and repulsive forces. The speed vector is computed with its dynamics equation and kinematics equation. However, pure APF (purely summing the attractive and repulsive forces) may not achieve desired cooperation in most cases. For example, if two robots detect a same target, both of them will track this target and therefore they will form a triangular pattern. This is not the optimal cooperation; the robot force is wasted because one of the robots can leave and search for other targets to maximize the number of observed targets.

A solution to avoid the disadvantage of pure APF is given a different weight to the attractive force and repulsive force for each robot. In this study, a weighted coefficient method based on Local Density of Robots and Targets (LDRT) is proposed to control the movement of robot in multi-target tracking. The basic idea is that if a robot locates at the region of great density of robots, it will be suffered more repulsive force and if it locates at the region of great density of targets, it will be received more attractive force. Consequently, robots could track more targets and coverage more area.

Figure 1 shows an example of control rule of a robot movement for multi-target tracking. Robot R1 and R2 are within the communication range of Robot Ri. So Ri experiences the virtual repulsive force due to R1 and R2. Target T1 and T2 are within the sensing range of Ri. T1 and T2 exert attractive force on Ri. Force vector F is the total force of repulsive and attractive. It can drive Ri move a new position at a speed. R3 is outside the communication range of Ri and T3 is outside the sensing range of Ri. So they have no effect to Ri. Robot movement is always far from other robots and it moves to the regions which have more targets.

LDRT algorithm: Every robot synchronously implements the same algorithm. In the process of every movement of robot, first, robot sends its coordinate and asks other robots which receive the message respond their current coordinates. After receives these messages, robot calculate the repulsive forces according to the relative position to it while robot senses targets. Robot calculates the attractive forces from targets. Robots are programmed to periodically calculate the forces that are acting on them, in order to choose a new path for motion. Repeating the process until the iteration ends.

When the robots are first switched ON, they have a zero velocity. If they experience any force, their velocity changes. Once the force on the robot is calculated, it is converted to a change in velocity of the robot. This is used to calculate the new velocity of the robot. Once the new velocity is computed, it is used to calculate the change in the displacement of the robot. The change in displacement gives the new position of the robot.

Assume $D_{R_i}^{Rob}$ is an estimate of the density of robots in a region normalized by the communication coverage area of a single robot at time t.

$$D_{R_i}^{Rob} = \frac{Num_{R_i}^{Rob}}{S_{R_i}} \tag{3}$$

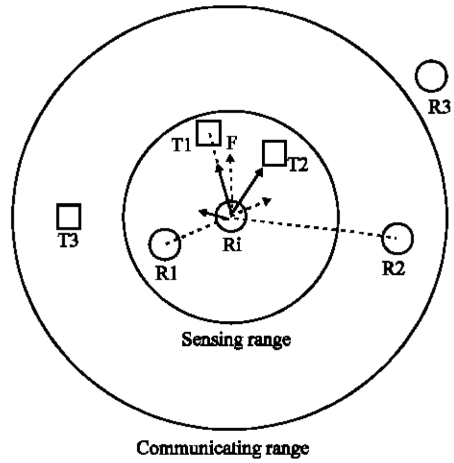

Fig. 1: An example of control rule of robot movement

Where, $Num_{R_i}^{Rob}$ is the number of robots within the communication range of R_i. S_{Ri} is the communication area of R_i.

Assume $D_{R_i}^{Tar}$ is an estimate of the density of targets in a region normalized by the sensor coverage area of a single robot at time t.

$$D_{R_i}^{Tar} = \frac{Num_{R_i}^{Tar}}{A_{R_i}} \tag{4}$$

Where, $Num_{R_i}^{Tar}$ is the number of targets within the sensor range of R_i. A_{ri} is the sensor area of R_i.

F_{ri} is the summation of the attractive and repulsive forces for R_i at time t; r_{Ri} is the set of detected targets of R_i; t_{Ri} is the set of the neighbor robots of R_i.

$$F_{R_i} = \sum_{j \in r_{R_i}} c_{R_i} \cdot F_{R_i,R_j} + \sum_{l \in t_{R_i}} c_{T_i} \cdot F_{R_i,T_l} \tag{5}$$

$$c_{R_i} = \begin{cases} \dfrac{D_{R_i}^{Rob}}{D_{R_i}^{Tar}} & D_{R_i}^{Tar} \neq 0 \\ 1 & D_{R_i}^{Tar} = 0 \end{cases} \tag{6}$$

$$c_{T_i} = \begin{cases} \dfrac{D_{R_i}^{Tar}}{D_{R_i}^{Rob}} & D_{R_i}^{Rob} \neq 0 \\ 1 & D_{R_i}^{Rob} = 0 \end{cases} \tag{7}$$

F_{Ri}, R_{Rj} is the repulsive force from neighbor robot R_j within the communication range of R_i at time t.

$$F_{R_i,R_j} = \sum_{l=0}^{r_{R_i}} \beta(|q_i - q_j| - cr)\frac{q_{ij}}{|q_i - q_j|} \tag{8}$$

$F_{Ri,Tl}$ is the attractive force to target T_l within the sensing range of R_i at time t.

$$F_{R_i,T_l} = \sum_{j=0}^{t_{R_i}} \alpha(|s_l - q_i| - sr)\frac{s_l - q_i}{|s_l - q_i|} \tag{9}$$

α, β is a positive gain constant, respectively.

According to corollary to Newton's Second Law of motion, the total force acting on a robot will cause it to accelerate, such that

$$F_{RI} = ma \tag{10}$$

Where, F_{Ri} is the force acting on the robot, m = 1 is the mass of the robot and a is the acceleration of the robot. a_{max} is the maximum acceleration of the robot. The robot is programmed with a timer that goes off every Δt sec and makes the robot recalculate the forces on it. If Δt is small, it can be used to approximate a_{max} in a 2D case:

$$F_{max} \approx \frac{v_{max}}{\Delta t} \tag{11}$$

Robots have a maximum acceleration and a maximum velocity. The forces acting on it have to be converted to a velocity and acceleration component at each point of the robot's motion. The acceleration on the robot is limited by limiting the maximum force that can act on the robots at any given time, while the velocity v is simply capped off after it exceeds v_{max} of the robot.

The state vector of robot is defined as

$$q_i = [x_i \quad \dot{x}_i \quad y_i \quad \dot{y}_i \quad \alpha_i \quad \dot{\alpha}_i]^T \tag{12}$$

And the drive force vector is defined as

$$u_i = [v_i^l \quad v_i^r]^T \tag{13}$$

Where, v_i^l is the line velocity and v_i^r is the rotational velocity. The dynamics equations are

$$\begin{aligned}
x_i(t+1) &= x_i(t) + \Delta t \cdot \dot{x}_i(t) \\
\dot{x}_i(t+1) &= v_i^l(t) \cdot \cos(\alpha_i(t) + \Delta t \cdot v_i^r(t)) \\
y_i(t+1) &= y_i(t) + \Delta t \cdot \dot{y}_i(t) \\
\dot{y}_i(t+1) &= v_i^l(t) \cdot \sin(\alpha_i(t) + \Delta t \cdot v_i^r(t)) \\
\alpha_i(t+1) &= \alpha_i(t) + \Delta t \cdot \dot{\alpha}_i(t) \\
\dot{\alpha}_i(t+1) &= v_i^r(t)
\end{aligned} \tag{14}$$

The details of the LDRT algorithm used by each robot are shown as follow:

(1)Initialization
    input $q_i(0)$;
    set sr;
    set cr;
    set t = 0;
    set t_{max};
While ($t<t_{max}$) do
{
(2) Calculate virtual forces
    Send (request, $q_i(t)$) within the communication range of R_i;
    Receive (response, q_j) within the communication range of R_i;
    Calculate $D_{R_i}^{Rob}$ according to formula (3);
    Calculate F_{R_i,R_j} according to formula (8);
    Detect the position of targets within the sensing range of R_i;
    Calculate $D_{R_i}^{Tar}$ according to formula (4);
    Calculate F_{R_i,T_l} according to formula (9);
    Calculate F_{R_i} according to formula (5);
(3) New position calculation
Calculate $q_i(t+1)$according to formula (11) and (14)
If the coordinate of $q_i(t+1)$ is outside the interest area, the robot stays at the nearest boundary;
(4)t = t+1;
}
End

RESULTS

We conducted experiments in simulation program to evaluate the effectiveness of the LDRT algorithm in addressing the multi-target problem. The program can be run under the combination of vary number of robots and targets. We compared the performance of the algorithm with other two algorithms: 1.APF (Artificial Potential Field); 2.SN (Static Node).

Simulation experiment: The simulation experiment is a 100×100 m square region with no obstacles and bounded plane. The number of robots is varied from 2 to 12. The radius of robot's sensors is varied from 10 m to 20 m and the radius of communication is varied from 20 m to 40 m. The maximum velocity of robot is 3 m sec^{-1}. The robot can not move out of the boundary of region.

The number of targets is varied from 2 to 12 also. The target is assumed having no functions of sensing environment. Every target moves linearly along its current direction of movement with uniform speed. While encountering the boundary of region targets move with mirror reflection. The maximum velocity is 2 m sec^{-1}.

At the beginning of each experiment, robots and targets are randomly positioned and oriented. Each experiment runs 1000 sec. $\Delta t = 1$ sec.

Five items of experiment are conducted. Each item tests 100 times with three tracking strategies. The average test results are the final results for performance analysis.

Figure 2 shows the snapshot of the experiment for four robot tracking four targets with LDRT algorithm. The small red hollow circles give the initial positions of the robots, while the small blue hollow squares give the initial positions of the targets. Traces of the robot and target paths are shown in gray solid lines and dashed lines respectively. The small red solid circles give the final positions of robots and the small blue solid squares give the final positions of targets. The gray circle around each robot represents the sensing range of that robot. Figure 2 shows the snapshot of screen after the simulation program runs 50 sec. In Fig. 2 R_0 is tracking T_2 and R_1 is tracking T_3. Because T_0 is not within the sensing range of R_2, R_2 moves randomly. R_3 is tracking T_1 and T_0.

Figure 3 is the results with the APF algorithm for multi-target tracking. R_0 is tracking T_2. R_1 and R_2 do not track any target. R_3 is tracking T_1 after random moving some time. In the figure, R_1 and R_3 initially move to opposite direction due to the repulsive force between them.

Figure 4 shows the snapshot with the SN algorithm.

In these figures, robots can track more targets using the LDRT algorithm and the APF algorithm. Only robots don't move using the SN algorithm. Because the algorithm adopts a motionless strategy for tracking, robots gain less times for tracking targets.

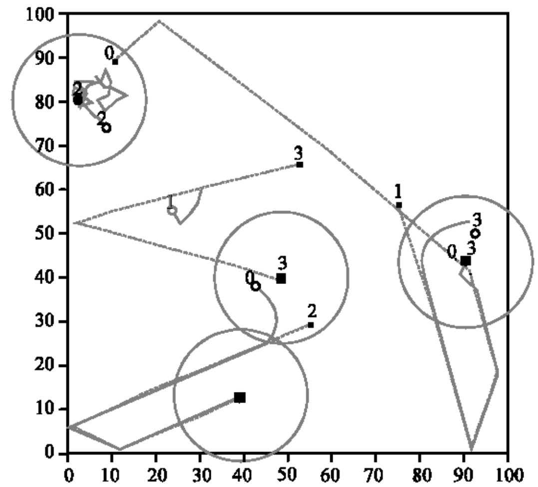

Fig. 2: The simulation result with LDRT

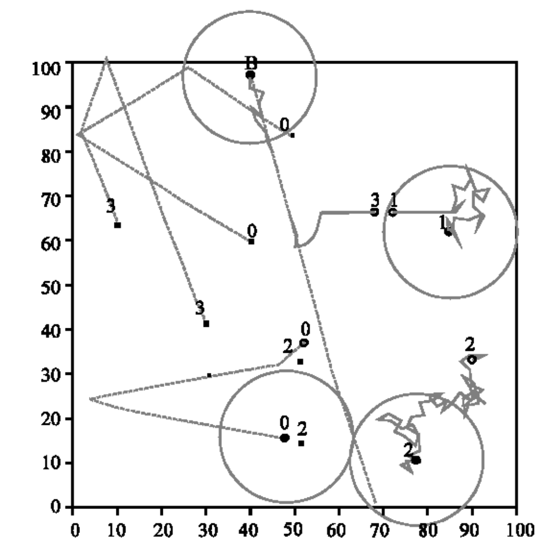

Fig. 3: The simulation result with APF

Performances comparison: Figure 5 shows the results of average observation for the ratio = 1 of the number of robots to the number of targets (n/m = 1) with three algorithms. Because robots with LDRT or APF move follow targets the average observation is very high at all times. As the virtual force exerting on robot is adjusted with the weighted factor of LDRT, robot will not lose target due to the influence of other robots and the observation is increased correspondingly. The total average observation approaches 88% using LDRT. And the total average observation is about 79% using APF. The performance of the LDRT algorithm is better than the APF algorithm.

The robots don't move with the SN algorithm. The less number of robots the low observation. When the number of robots and the coverage of region are increased the observation is improved. While the number of robots is 2, the average observation is 11.4%. When the number of robots reaches 12, the average observation is 56.4%.

Figure 6 shows the results of average coverage for n/m=1 with three different algorithms. The average coverage varies from about 12 to 60%. LDRT and APF are the same basically in average coverage. SN is a little low. The reason is that the coverage area is increased as the movement of robots with the former two algorithms and the coverage is invariable as the robot is stationary with SN.

Figure 7 gives the comparison of average observation for the number of robots is 6, the number of targets is varied from 2 to 12 with the three algorithms. The change trends of observation are the same. The LDRT is best and the SN is worst. The best observation is 89.1% when the number of targets is 4 with LDRT. And the best observation is 75.9% when the number of targets is 6 with APF. The observation of LDRT and APF is almost the

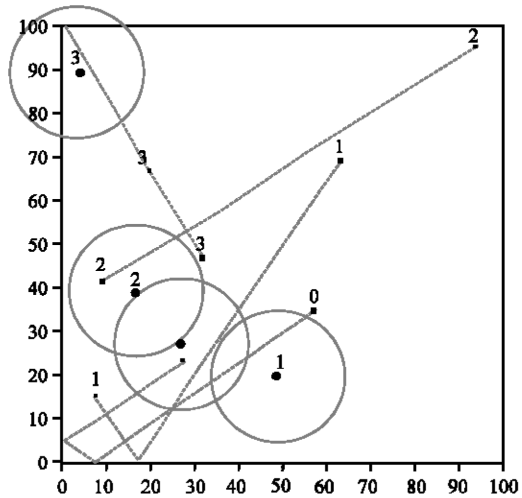

Fig. 4: The simulation result with SN

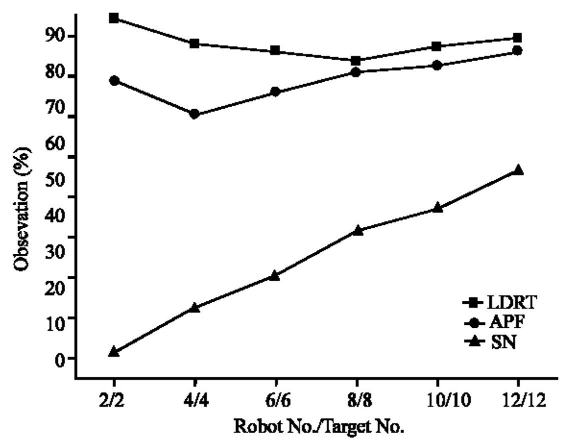

Fig. 5: The comparison of observation with n/m=1

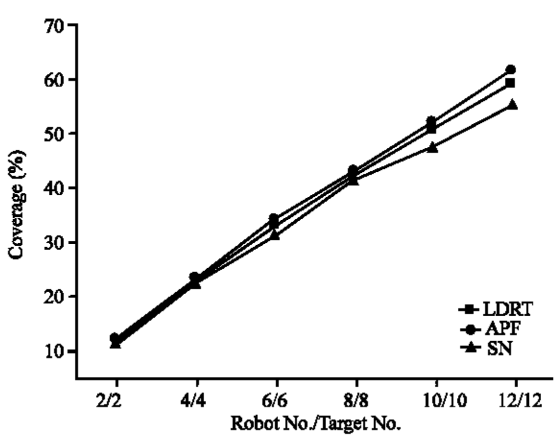

Fig. 6: The comparison of coverage with n/m = 1

same when the number of targets is 10 and 12. The worst observation is 17% when the number of targets is 12 with SN.

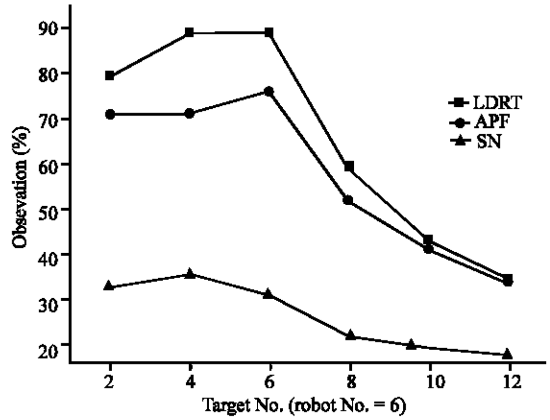

Fig. 7: The comparison of observation with n = 6

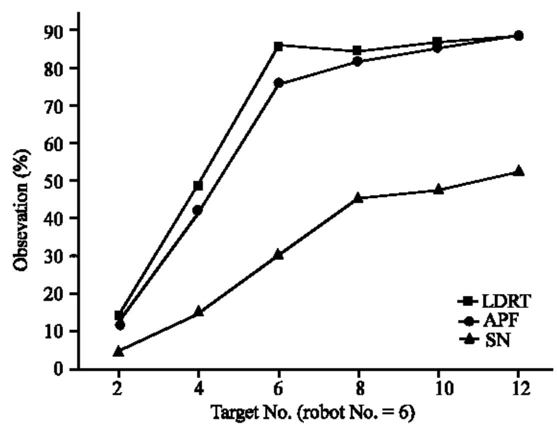

Fig. 8: The comparison of observation with m = 6

Figure 8 shows the comparison of average observation for the number of targets is 6 when the number of robots varied from 2 to 12 with the three algorithms. The change trends of observation are the same with the number of robots increasing. The LDRT algorithm performs better than the APF algorithm. The best observation is 88.5% when the number of robots is 12 with LDRT. And the best observation also is 88.5% when the number of robots is 12 with APF. The observation of LDRT and APF is almost the same when the number of robots is 10. Consequently, when the number of robots increases certain numbers the two algorithms have little difference. But the best observation is only 52.4% when the number of robots is 12 with SN.

Figure 9 shows the comparison of average observation for all the number of targets and robots are 6, the radius of sensing varied from 10 to 20 m and the radius of communication varied from 20 to 40 m with the three algorithms. The experiments are conducted in five groups. The observation is increased as the increasing of

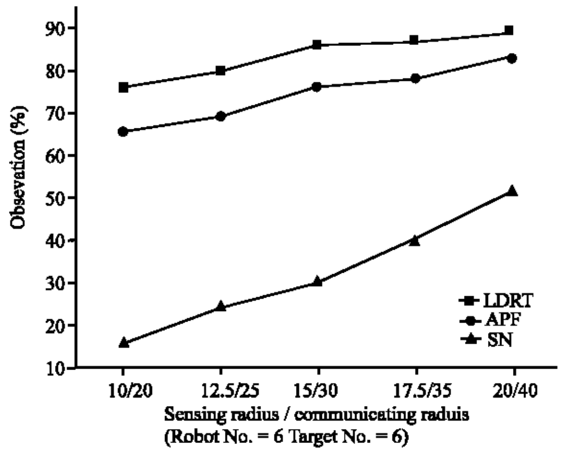

Fig. 9: The comparison of observation with vary of sensing range and communicating range

communication radius of robots with the three algorithms. The performance is improved approximately at the percent with LDRT and APF. When the radius of sensing and the radius of communication is increased from 10/20 m to 20/40 m, the observation is improved from 75.9 to 89.1% with LDRT. And the observation is also improved very much with SN.

The performance of observation is improved utilizing the local density of robots and targets as weighted factor compare with fixed weighted factor for multi-target tracking. The reason is that the weighted factor of virtual force is changed as varies of the number of robots and targets in one region and the factor can overcome the shortcoming of APF. The LDRT algorithm makes robots self-organizing coordinate for the task of multi-target tracking.

CONCLUSIONS

Many real-world applications in security, surveillance and reconnaissance tasks require multiple targets to be monitored using mobile sensors. We propose a novel distributed algorithm for the solution of multi-target tracking. The algorithm is an improved artificial potential field algorithm based on local density of robots and targets in one region. This density is as a weighted factor for adjusting the virtual force to control robot move. The simulation verified the validity of the LDRT algorithm. The cooperation among robots was controlled by the algorithm and the algorithm performs better than other two algorithms. The key issue of this research is to use the algorithm in real experiments of multi-target tracking in the future.

REFERENCES

Howard, A., M.J. Mataric and G.S. Sukhatme, 2002. Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem. 6th International Symposium on Distributed Autonomous Robotic Systems (DARS), pp: 299-308.

Jung, B. and G.S. Sukhatme, 2004. A generalized region-based approach for multitarget tracking in outdoor environments. In IEEE International Conference on Robotics and Automation, New Orleans, LA., pp: 2189-2195.

Jung, B. and G.S. Sukhatme, 2006. Cooperative Multi-robot Target Tracking. The 8th International Symposium on Distributed Autonomous Robotic Systems. pp: 81-90.

Li, D., K. Wong, Y.H. Hu and A. Sayeed, 2002. Detection, Classification and Tracking of Targets in Distributed Sensor Networks. IEEE Signal Processing Maga., 19: 17-29.

Li, T.H.S., S. Chang and W. Tong, 2004. Fuzzy Target Tracking Control of Autonomous Mobile Robots by Using Infrared Sensors. IEEE Trans. Fuzzy Syst., 12: 491-501.

Murrieta-Cid, R., H´. Gonz´alez-Ba˜nos and B. Tovar, 2002. A reactive motion planner to maintain visibility of unpredictable targets. In Proceeding of IEEE International Conference on Robotics and Automation, pp: 4242-4247.

Parker, L., 1999. Cooperative robotics for multi-target observation. Intelligent Automation Soft Compu., 5: 5-19.

Parker, L., 2002. Distributed algorithms for multi-robot observation of multiple moving targets. Autonomous Robots, 12: 231-255.

Spletzer, J. and C. Taylor, 2003. Dynamic sensor planning and control for optimally tracking targets. Int. J. Robotics Res., 22: 7-20.

Spears, W.M. and D.F. Gordon, 1999. Using artificial physics to control agents. In Proceedings of IEEE International Conference on Information, Intelligence and Systems.

Stroupe, A. and T. Balch, 2003. Value-Based Observation with Robot Teams (VBORT) using probabilistic techniques. In Proceedings of the International Conference on Advanced Robotics, Coimbra, Portugal.