Distributed Target Tracking Using Binary Sensors with Imprecise Range Measurements

Zijian Wang, Eyuphan Bulut, and Boleslaw K. Szymanski

Department of Computer Science and Center for Pervasive Computing and Networking, RPI, Troy, NY

Abstract—**We study target tracking with wireless binary sensor networks, in which each sensor can return only 1-bit information regarding target's presence or absence in its sending range. A novel, real-time and distributed target tracking algorithm for an imperfect binary sensing model is proposed, which is an extension of our previous work on the ideal binary sensing model.**

I. INTRODUCTION

Target tracking is a representative and important application for wireless sensor networks. One of the fundamental studies of target tracking focuses on binary sensing that provides just one bit of information about the target: whether it is present within the sensing range or not. There are two kinds of binary sensing model for binary sensor networks. In the ideal binary sensing model, each node can detect exactly if the target falls into its sensing range R. In an imperfect binary sensing model, the target is always detected within an inner disk of radius R_{in} but is detected only with some nonzero probability in an annulus between the inner disk and an outer disk of radius R_{out}. Targets outside the outer disk are never detected.

In this paper, we extend our previous work and propose a distributed target tracking algorithm that can be used for an imperfect binary sensing model while keeping all the other properties of its predecessor.

II. TRACKING ALGORITHM

We take a worst-case approach to the information provided by the imperfect binary sensing model: if a sensor output is "1", then we assume that the target is somewhere inside the large disk of radius R_{out}; if a sensor output is "0", then we assume that the target is somewhere outside the small disk of radius R_{in}. We cannot identify circular arcs that the target crosses under imperfect binary sensing model, which was possible in the ideal binary sensing model [1]. We can only identify that the target must be within the ring determined by R_{in} and R_{out}. However, we can use a thin ring section which is determined by the neighbor outputs to approximate the circular arcs and then to estimate the position of the target.

A. Initialization

In the initialization procedure, each node establishes a list of

Zijian Wang, Eyuphan Bulut and Boleslaw Szymanski are with the Department of Computer Science and the Center for Pervasive Computing and Networking, Rensselaer Polytechnic Institute, Troy, NY 12180 USA. (e-mail: {wangz, bulute, szymansk}@cs.rpi.edu).

its neighbors and calculates the exact angle corresponding to a neighbor depending on the output and the relative position of that neighbor. The three instances for neighbor (node Y) that outputs bit "1" are shown in Fig. 1. If node Y outputs bit "1", we can only be sure that the target is within sensing range R_{out}. When node X senses there is a change in the status of the target, it knows that the target is within the ring determined by R_{in} and R_{out}. Depending on the relative position of node Y to node X, there would be up to two angles corresponding to node Y's range circle defined by the intersection of R_{in} circle of node X and R_{out} circle of node Y. If there are two such angles, we choose the one that ensures that the target falls within it; for example we choose $\angle b_1 o b_2$ and $\angle a_1 o a_2$ in Fig. 1 (a) and (b) as the angles corresponding to neighbor Y. If there exists only one angle for node Y, then it is the corresponding angle, as shown in Fig. 1 (c). The three instances for neighbor (node Y) that outputs bit "0" are shown in Fig. 2. If there are two such angles for node Y, we choose the one that ensures that the target is outside of it; for example we choose $\angle a_1 o a_2$ and $\angle b_1 o b_2$ in Fig. 2 (a) and (b) as the angles corresponding to neighbor Y. For the instance shown in Fig. 1(c), we can not determine that the target is outside $\angle a_1 o a_2$ because the target could be within $\angle a_1 o a_2$ no matter what node X output is.
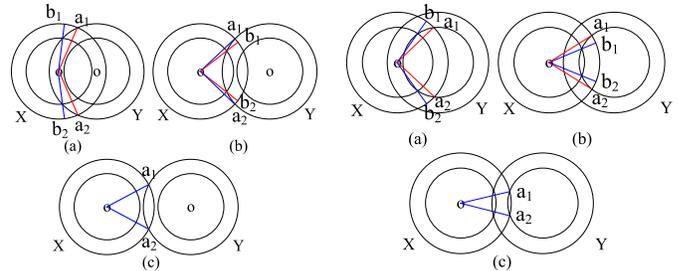

Fig. 1 An angle defined by the neighbor's output "1"

Fig. 2 An angle defined by the neighbor's output "0"

B. Location Estimate

At the moment at which the node discovers the change in the target's presence, it calculates the final angle corresponding to the ring section that the target is crossing using the same angle combination method as in the ideal binary sensing model [1]. Then, the thickness of the ring section is recalculated to make the estimation of target position more accurate. We calculate the intersection points of each pair of node X's neighbors that output bit "1". The intersection point that falls into the final angle and is furthest away from the center of node X determines one of the boundaries of the ring section. A new thinner ring

section is determined by this intersection point, R_{out} circle and the final angle, an example of the resulting ring section "abcd" is shown in Fig. 3.
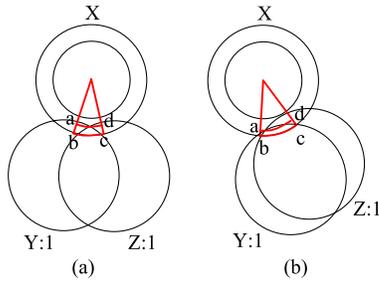

(a)   (b)
Fig. 3 Ring section thickness calculation

C. Velocity and Trajectory Estimate

We use the same method as in [1] to estimate target velocity and trajectory but change the weight of each estimate in trajectory estimate as: w=|ring|/|ring section|, where |ring| is the area of the ring determined by circles R_{in} and R_{out} and
|ring section| is the area of corresponding ring section whose middle point is the estimated target location.

III. SIMULATION

A. Location Estimate

We kept the number of nodes fixed at 800 within an 800 by 800 square units area and varied the sensing range R_{out} from 40 to 150 units. Three types of trajectories have been considered: (1) linear, (2) circular, and (3) a piece-wise linear trajectory with random turns. As in [2], we set R_{in} =0.9*R_{out}. We use a constant distribution detection probability: P_{detect}=(R_{out} –d)/(R_{out} – R_{in}) when R_{in}<d<R_{out}, where d is the distance between the sensing node and the target. We compare our algorithm with the following other four algorithms introduced in [2] and [3]: (1) Equal Weight; (2) Distance Weight; (3) Duration Weight; and (4) Line Fit.
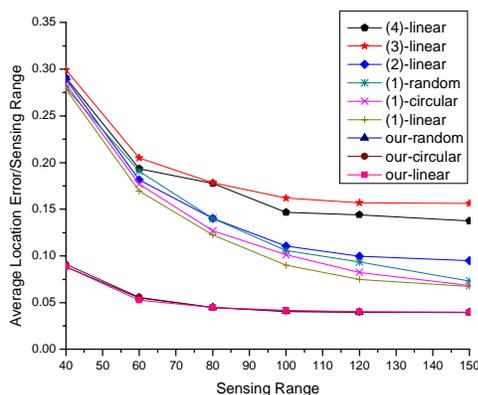

Fig. 4 The location estimate accuracy

Fig. 4 shows the location estimate accuracy results that demonstrate that our algorithm gets the best results of all compared algorithms. Additionally, the location estimate accuracies yielded by our algorithm for each of the three trajectories are nearly the same, which demonstrates that our algorithm works well for all kinds of trajectories.

B. Velocity Estimate

We tested the accuracy of velocity estimation under the configuration of 800 nodes with R_{out}=40 unit and R_{in}=0.9*R_{out} unit sensing ranges using the constant distribution detection probability. The target moves along a linear trajectory with the velocity changing suddenly to a random value that is a multiple of R_{out}/15 unit/second several times during simulation. Fig. 5 shows the estimated versus real velocities as a function of time. Clearly, these two agree well, although there is some delay before the change of real velocity is reflected in its estimate.
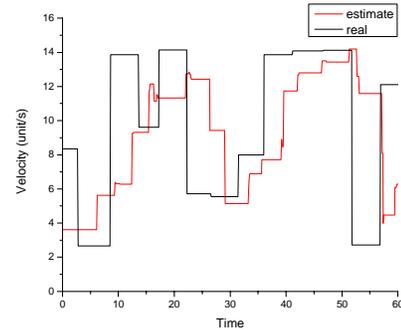

Fig. 5 The estimated velocity versus the real velocity

C. Trajectory Estimate

We estimate trajectory under the same configuration as used in velocity estimate. We measure the accuracy of estimated trajectory using the average difference between the estimated and real trajectories. It is calculated using the area of a polygon formed by these two trajectories divided by the length of the real target trajectory. The average accuracies are 0.287, 1.811 and 2.873, for linear, circular and piece-wise linear trajectories with random turns, respectively.

IV. CONCLUSION

In this paper, we extend our study of target tracking problem under the ideal binary sensing model and we introduce a real-time distributed target tracking algorithm without time synchronization for imperfect binary sensing. Extensive simulations of this algorithm performed under different configurations and scenarios are reported. We observe that our algorithm yields good performance and outperforms other algorithms in terms of accuracy of its estimates of the target location, velocity and trajectory.

REFERENCES

[1] Z. Wang, E. Bulut, and B. K. Szymanski, "A distributed cooperative target tracking with binary sensor networks," Proc. IEEE International Conference on Communication Workshops, May 2008, Beijing, China, pp. 306-310.
[2] K. Mechitov, S. Sundresh, Y. Kwon, and G. Agha, "Cooperative tracking with binary-detection sensor networks," Technical Report UIUCDCS-R-2003-2379, University of Illinois at Urbana-Champaign, September 2003.
[3] W. Kim, K. Mechitov, J.-Y. Choi, and S. Ham, "On target tracking with binary proximity sensors," Proc. IPSN, 2005.