A Greedy Strategy for Tracking a Locally Predictable Target among Obstacles

Tirthankar Bandyopadhyay*, Yuanping Li*, Marcelo H. Ang Jr.*, and David Hsu†

*Department of Mechanical Engineering
National University of Singapore
Singapore, 119260, Singapore

†Department of Computer Science
National University of Singapore
Singapore, 117543, Singapore

***Abstract*— Target tracking among obstacles is an interesting class of motion planning problems that combine the usual motion constraints with robot sensors' visibility constraints. In this paper, we introduce the notion of *vantage time* and use it to formulate a risk function that evaluates the robot's advantage in maintaining the visibility constraint against the target. Local minimization of the risk function leads to a greedy tracking strategy. We also use simple velocity prediction on the target to further improve tracking performance. We compared our new strategy with earlier work in extensive simulation experiments and obtained much improved results.**

I. Introduction

The target tracking problem considers motion strategies for an autonomous mobile robot to track a moving target among obstacles, *i.e.*, to keep the target within the robot sensor's visibility region. Target tracking has many applications. In home care settings, a tracking robot can follow elderly people around and alert caregivers of emergencies. In security and surveillance systems, tracking strategies enable mobile sensors to monitor moving targets in cluttered environments.

Target tracking is an especially interesting class of motion planning problems. Just as in classic motion planning [7], we must consider *motion constraints* resulting from both obstacles in the environment and the robot's mechanical limitations. In particular, the robot must not collide with obstacles. Target tracking has the additional *visibility constraints* due to sensor limitations, *e.g.*, obstacles blocking the view of the robot's camera. The robot must move in such a way that the target remains visible at all times. Both motion constraints and visibility constraints play important roles in target tracking.

Inspired by earlier work [5], we propose a greedy strategy for target tracking. It uses the robot's sensor to acquire local information on the target and the environment, and use this information to compute the robot's motion at each step. Thus, it does not need *a priori* knowledge of the environment or localization with respect to (w.r.t.) a global map. The key element of the greedy strategy is a local function φ that gives a combined estimate of the immediate risk of losing the target and the future risk. Our definition of φ is based on two important considerations:

- Vantage time, which is a combined estimate of the robot's ability to maneuver against the target in both the current and future time.
- The target's instantaneous velocity, which can be estimated locally, indicates the target's future movement.

We show that by leveraging these two considerations, our new strategy achieves significant improvement in performance over the earlier strategy presented in [5].

In the following, after a brief review of related work (Section II), we state the target tracking problem formally (Section III) and present our solution (Section IV). We have implemented the new tracking strategy and compared it with an existing one in simulated environments. The simulation results are shown in Section V. Finally, we conclude with some remarks on future research directions (Section VI).

II. Related Work

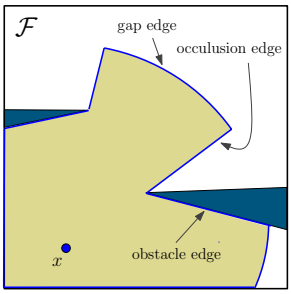
Tracking strategies differ greatly, depending on whether the environment is known in advance. If both the environment and the target trajectory are completely known, optimal tracking strategies can be computed by dynamic programming [8] or by piecing together certain canonical curves [3], though usually at a high computational cost. If only the environment is known, one can preprocess the environment by decomposing it into cells separated by critical curves. The decomposition helps to identify the best robot action as well as to decide the feasibility of tracking [11]. Often, neither the environment nor the target trajectory is known in advance. One approach in this case is to move the robot so as to minimize an objective function that depends on the shortest distance for the target to escape from the visibility region of the robot's sensor, abbreviated as SDE [5], [9], [12]. Our work belongs to this category. An important issue here is to balance the immediate risk against the future risk of losing the target.

The earlier work heavily replies on SDE in formulating the risk function. In contrast, the concept of vantage time introduced here provides a more systematic way to integrate various factors contributing to the escape risk and derive a risk function offering better tracking performance. The use of local velocity estimation to predict target motion further improves the performance of our tracking strategy.

Target tracking has also been studied jointly with other objectives, such as stealth [1], which requires the tracking robot to remain "invisible" to the target, and robot localization [4].

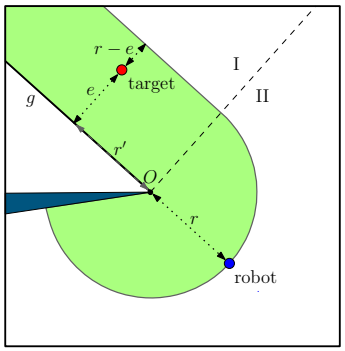
A problem that is related, but complementary to that of target tracking is covert path planning [2], [10], which tries to minimize the robot's exposure to observers with known or partially known locations.

(a)



(b)

Fig. 1. A robot tracking a target in a planar environment. (*a*) A robot mounted with a laser range finder tracks another robot. Courtesy of H.H. González-Baños. (*b*) The visibility set (shaded) for the robot located at x



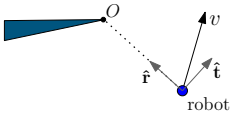


Fig. 2. The robot's vantage zone (shaded) w.r.t. a gap edge g. In the robot's visibility set, region I lies above the dashed line, and region II lies below.

III. Problem Formulation

The goal of target tracking is to keep the target within the visibility region of the robot's sensor at all times. The robot and the target are assumed to operate in a planar environment populated with obstacles (Fig. 1) For simplicity, both are modeled as points in the plane. The extension to the usual cylindrical robots is straightforward. We assume that the robot has no prior knowledge of the environment, but is equipped visual sensors (*e.g.*, cameras and laser range-finders) to acquire information on the local environment and identify the target.

We use the standard straight-line visibility model for the robot's sensor. Let $\mathcal{F}$ denote the subset of the plane not occupied by obstacles. The target is *visible* to the robot if the line of sight between them is free of obstacles, and the distance between them is smaller than $D_{\max}$, the maximum sensor range. The *visibility set* $\mathcal{V}(x)$ of the robot at point x consists of all the points at which the target is visible (Fig. 1*b*):

$$\mathcal{V}(x) = \{x' \in \mathcal{F} \mid \overline{xx'} \subset \mathcal{F} \text{ and } d(x,x') \leq D_{\max}\},$$

where $d(x,x')$ denotes the distance between x and x'. If the robot must monitor the target from a minimum distance away, we can impose the additional constraint $d(x,x') \geq D_{\min}$. In the following, the visibility set is always taken w.r.t. the current robot position, and so we omit the argument x.

The robot's motion is modeled with a simple discrete-time transition equation. Let $x(t)$ denote the position of the robot at time t. If it chooses a velocity $v(t)$ at time t, its new position $x(t+1)$ after a fixed time interval Δt is given by

$$x(t+1) = x(t) + v(t)\Delta t.$$

Here, we implicitly assume that sensing occurs every Δt time. This discrete model is effective as long as Δt remains small. As we will see, our tracking strategy is very efficient. Based on the experience of previous work [5], we expect it to run at the rate of 10 Hz, sufficient for keeping Δt small in many common tasks. The robot has velocity bound V, but has no other kinematic or dynamic constraints. So, in one time step, it can reach anywhere inside a circle with center $x(t)$ and radius $V\Delta t$, unless it is obstructed by obstacles.

The target's motion is modeled similarly, but has velocity bound V'. Typically, $V \geq V'$. Otherwise, the target can easily escape by running straight ahead with maximum velocity, and the tracking problem is uninteresting. We further assume that at any time, the robot has an estimate of the target velocity.

We now state the problem formally:

Problem 1: A tracking strategy computes a sequence of actions, namely, the velocities $v(t), t = 0, 1, \ldots$, for the robot so that at any time t, the target position lies inside $\mathcal{V}$.

IV. The Greedy Tracking Strategy

Our main idea is to define a function φ that estimates the risk for the target to escape from $\mathcal{V}$. The visibility set $\mathcal{V}$ can be approximated by a generalized polygon with linear and circular boundary edges. The boundary of the polygon consists of *gap edges* and *obstacle edges* (Fig. 1*b*). Obstacle edges come from parts of the obstacle boundaries that block the robot's line of sight. Gap edges lie inside $\mathcal{F}$. They are further divided into two types: *occlusion edges* that result from occlusion by the obstacles and *range edges* that result from maximum or minimum sensor range limits. Both obstacle and occlusion edges are linear. Range edges are circular. To escape from $\mathcal{V}$, the target must exit one of the gap edges. Below, we first define the escape risk w.r.t. a single gap edge and then combine them to form the overall risk function φ. The robot then chooses its action, *i.e.*, the velocity v, to minimize φ.

A. Escape Risk with respect to a Single Gap Edge

Let us first consider the more interesting type of gap edges, occlusion edges. Suppose that g is an occlusion edge. Let ℓ denote the visibility line containing g, and let O denote the obstacle vertex abutting ℓ (Fig. 2). We call O the *occlusion point*. The risk φ_g of escaping through g depends on several quantities. Clearly, φ_g depends on e, the shortest distance from the target to g. The smaller the e is, the easier it is to escape through g. In a subtle way, φ_g also depends on the robot's and the target's relative positions w.r.t. O: specifically, r, which is the distance from the robot to O, and r', the projected distance from the target to O along ℓ. If r is much smaller than r', the robot can move the gap edge g away from the target much faster by rotating ℓ around O.

To combine e, r, and r' into a single risk estimate, we introduce the notion of *vantage zone*. The robot's vantage zone $\mathcal{D}(g)$ w.r.t. a gap edge g is the set of points that are closer to

g than the robot is (Fig. 2). Again, we omit the argument g from $\mathcal{D}(g)$ when it is clear from the context which gap edge is involved. Geometrically, $\mathcal{D}$ is a band adjacent to g with width r. It is so named, because if the target is outside $\mathcal{D}$, then the robot can always reach g before the target and prevent it from escaping through g by running towards O, the closest point in g from the robot[1]. Thus, the robot should keep the target out of $\mathcal{D}$, and a good estimate of the target's escape risk is the amount of time $t_{\mathcal{D}}$ that the targets needs to reach the boundary of $\mathcal{D}$. We call $t_{\mathcal{D}}$ the *vantage time*. If the target is inside $\mathcal{D}$, $t_{\mathcal{D}}$ is positive by convention and indicates the amount of time that the robot needs to push the target out of $\mathcal{D}$. If the target is outside $\mathcal{D}$, $t_{\mathcal{D}}$ is negative and indicates the amount of time that the robot can keep the target away from $\mathcal{D}$.

The robot's velocity v can be decomposed into a radial component v_{r} towards O and a tangential component v_{t} perpendicular to v_{r} (Fig. 2). The tangential component v_{t} causes the robot to swing out and immediately increases e, thus reducing the current escape risk. The radial component v_{r} causes the robot to approach O. It does not affect e immediately. Instead, it decreases r so that future tangential motion will increase e much faster. In this sense, v_{r} reduces future escape risk. Since the robot's velocity is upper bounded, we must choose v_{r} and v_{t} carefully to balance the current and the future risk. For this, let us examine the effects of v_{r} and v_{t} on $\mathcal{D}$. Fig. 2 shows that v_{r} shrinks the width r of $\mathcal{D}$ at the rate

$$dr/dt = v_{\mathrm{r}}, \tag{1}$$

and that v_{t} rotates $\mathcal{D}$ about O with angular velocity

$$\omega = v_{\mathrm{t}}/r. \tag{2}$$

Both components can be used to keep the target out of $\mathcal{D}$ by reducing $t_{\mathcal{D}}$, but their effectiveness depends on the target position w.r.t. to g. There are two cases.

a) Case I: The closest point in g to the target is interior to g. This case corresponds to region I marked in Fig. 2. The vantage time $t_{\mathcal{D}}$ depends on $r - e$, the distance between the target and the boundary of $\mathcal{D}$. The rate of change of $r - e$ is given by

$$d(r-e)/dt = dr/dt - de/dt. \tag{3}$$

Substituting (1) and (2) into (3), we have

$$\begin{aligned} d(r-e)/dt &= v_{\mathrm{r}} - (v'_{\mathrm{e}} - r'\omega) \\ &= v_{\mathrm{r}} + (r'/r)v_{\mathrm{t}} - v'_{\mathrm{e}}, \end{aligned} \tag{4}$$

where v'_{e} is the effective escape velocity, *i.e.*, the target velocity v' projected in the direction perpendicular to g. To compute $t_{\mathcal{D}}$ exactly, we must integrate (4) and solve the integral equation

$$r_0 - e_0 = \int_0^{t_{\mathcal{D}}} (v_{\mathrm{r}} + (r'/r)v_{\mathrm{t}} - v'_{\mathrm{e}})\, dt,$$

where r_0 and e_0 denote the values of r and e at current time step. This is not possible, as we do not know the future target actions. Instead, we estimate the target escape risk φ_g by approximating $t_{\mathcal{D}}$ using only information available at the current time step:

$$\varphi_g = \frac{r_0 - e_0}{v_{\mathrm{r}} + v_{\mathrm{t}}(r'_0/r_0) - v'_{\mathrm{e}}}, \tag{5}$$

where r'_0 is the value of r' at the current time step. Now, for a single gap edge g, target tracking reduces to minimizing the risk function $\varphi_g(v_{\mathrm{r}}, v_{\mathrm{t}})$. We do this by differentiating φ_g and computing its negated gradient:

$$-\nabla\varphi_g = \frac{\varphi_g}{v_{\mathrm{eff}}}\left(\frac{r'_0}{r_0}\hat{\mathbf{t}} + \hat{\mathbf{r}}\right). \tag{6}$$

In (6), $\hat{\mathbf{t}}$ and $\hat{\mathbf{r}}$ are unit vectors in the tangential and radial directions, respectively, and $v_{\mathrm{eff}} = v_{\mathrm{r}} + v_{\mathrm{t}}(r'_0/r_0) - v'_{\mathrm{e}}$ is the effective velocity in the direction along the shortest path from the target to g. In this case, v_{eff} is perpendicular to g.

The robot's action v w.r.t. g is simply $-\nabla\varphi_g$. Eq. (6) shows that the direction of v is $(1/\sqrt{r_0^2 + {r'_0}^2})(r'_0\hat{\mathbf{t}} + r_0\hat{\mathbf{r}})$. It depends only on r_0 and r'_0, which, intuitively, measure the robot's and the target's abilities to swing the visibility line ℓ against each other. When r_0 is smaller than r'_0, swinging is effective. Thus, the tangential component gets higher weight. When r_0 is larger than r'_0, the opposite holds. The magnitude of v acts as a weight when there are multiple gap edges. It depends on all three quantities, r, r', and e. In particular, when e is small w.r.t. to a gap edge g, φ_g becomes large. Thus, $-\nabla\varphi_g$ becomes large according to (6), and the corresponding action gets higher weight (see next subsection).

b) Case II: The closest point in g to the target is an endpoint of g. This case corresponds to region II marked in Fig. 2, and the closest point in g to the target is the occlusion point O. Here, the tangential component v_{t} has no effect on $t_{\mathcal{D}}$. Thus,

$$\varphi_g = \frac{r_0 - e_0}{v_{\mathrm{r}} - v'_{\mathrm{e}}}, \tag{7}$$

which can be obtained from (5) by setting $r' = 0$. The corresponding gradient of φ_g is then

$$-\nabla\varphi_g = \frac{\varphi_g}{v_{\mathrm{eff}}}\hat{\mathbf{r}}, \tag{8}$$

where v_{eff} is still the effective velocity in the direction along the shortest path from the target to g, but this time, it is directed towards O and is equal to $v_{\mathrm{r}} - v'_{\mathrm{e}}$.

Together, Eqs. (6) and (8) reveal that the robot's action is continuous over the entire domain, provided that so is the target's action. This is an important advantage in practice.

Let us now turn to range edges. For lack of space, the detailed derivation is omitted. We redefine O as the closest point in a range edge to the target position. Since r and r' are defined w.r.t. to O, their definitions change accordingly, but the definition of e remains the same as before. With these, the risk function and the resulting robot action are the same as those in (7) and (8).

B. Escape Risk with respect to All Gap Edges

A visibility set may contain many gap edges. Based on the target's motion patterns, we can identify the important ones and improve tracking performance. Let the *heading probability*

[1]Recall the assumption that the robot's velocity bound is greater than that of the target.

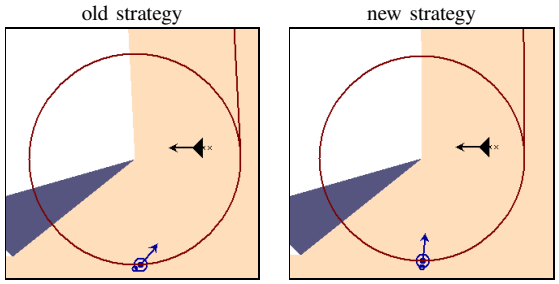


Fig. 3. A scenario in which swinging provides little advantage. In this and all following examples, the shaded region indicates the visibility set w.r.t. the current robot position. A small blue circle marks the robot position. A filled black triangle mark the target position. The associated arrows indicate the robot's and target's velocity directions. When shown, the robot trajectory is marked by blue circles. A circle filled with cyan indicates that the target it not visible at that time step. The target trajectory is marked by black crosses.

p_g be the probability of the target headed to a gap edge g. The overall risk φ is then computed as the expected risk over all the gap edges:

$$\varphi = \sum_g p_g \varphi_g. \tag{9}$$

Recall that φ_g is an estimate of the vantage time $t_{\mathcal{D}}$, and thus φ is the expected vantage time over all the gap edges. To find the robot action, we solve a simple optimization problem:

$$\min_{v_\mathrm{r}, v_\mathrm{t}} \varphi(v_\mathrm{r}, v_\mathrm{t}) \quad \text{subject to} \quad {v_\mathrm{r}}^2 + {v_\mathrm{t}}^2 = V^2. \tag{10}$$

We solve (10) by computing the negated gradient of φ:

$$-\nabla\varphi = -\nabla(\sum_g p_g \varphi_g) = -\sum_g p_g \nabla\varphi_g, \tag{11}$$

which is then scaled to magnitude V to give the action v for the robot. Eq. 11 shows that the overall action for the robot is a linear combination of the actions w.r.t. the individual gap edges, weighted by the heading probabilities.

The benefits of the new risk function is best illustrated in comparison with a related risk function introduced in earlier work [5]. For occlusion edges, the old risk function is a monotonic function of the ratio r/e and completely ignores r'. To simplify the presentation, we assume that the robot and the target have the same velocity bounds in all the following examples.

Consider the scenario in Fig. 3. The target is closer to the gap edge than the robot, and is moving towards the gap edge. The robot will lose the target unless it runs straight towards the occlusion point. Since r' is small, swinging in the tangential direction provides little advantage to the robot. Since the old risk function ignores r', it fails to recognize this and generates roughly equal amount of motion in the tangential and the radial directions. The new risk function puts almost all the motion in the radial direction towards the occlusion point.

Now consider another scenario (Fig. 4). The target is very close the gap edge, and thus e is small. As a result, the old risk function becomes very large and generates almost a full swing for the robot in order to reduce the current escape risk. However, the situation is in fact not that critical at all. The target is still a small distance away from the gap edge, leaving some time for maneuvering. More importantly, the robot is slightly closer to the occlusion point than the target. A small swing is sufficient to keep the target visible. Too much swing in the tangential direction reduces the motion in the radial direction and increases the future escape risk. This eventually causes the old risk function to lose the target. The new risk handles this situation much better. It always keeps the target visible until it "eliminates" the gap edge in the end (Fig. 4).

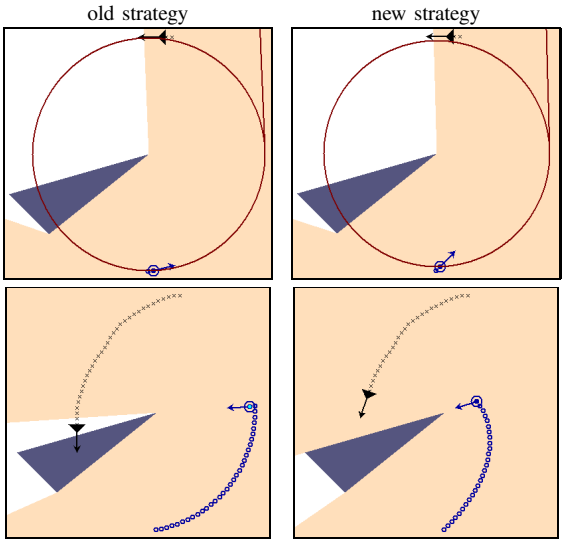


Fig. 4. A scenario in which too much swinging increases future risk.

These two scenarios show conclusively that along with r and e, the ratio (r'/r), which measures the robot's and the target's relative positions w.r.t. to the occlusion, plays an important role in risk estimation. This is one major reason why the new risk function performs better.

C. Heading Probability Estimation

To estimate heading probabilities, we need the current target velocity v'. At any time, we maintain an estimate of v' by storing a short history of the target trajectory and extrapolating. Many other methods for velocity estimation are possible. For simplicity, the uncertainty in estimating the direction θ of v' is assumed to follow a Gaussian distribution $f(\theta)$. The variance of the Gaussian indicates our confidence in estimating the target behavior. Other distributions, even non-parametric ones, can be used instead of the Gaussian, depending on the method of velocity estimation. Our method for computing heading probabilities is general and works with any distribution.

It is natural to assume that the target will exit a gap g, if it is headed to g. In other words, suppose that λ_θ is the ray originating from the current target position and having direction θ. The target will exit g if λ_θ intersects g. Thus, p_g can be estimated from the target's estimated velocity distribution and the angle subtended by g:

$$P(g) = \int_{\Theta_g} f(\theta)\, d\theta,$$

where θ lies in the angular range Θ_g if and only if λ_θ intersects g (Fig. 5). This seems reasonable, unless we consider a gap

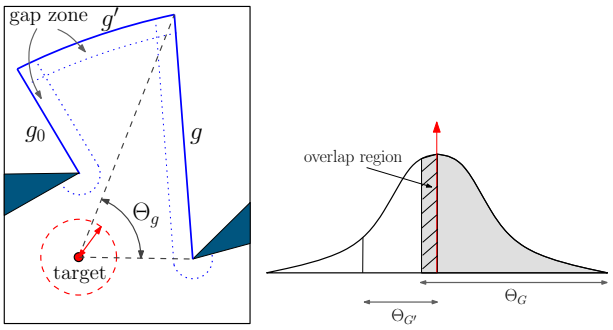


Fig. 5. Estimating heading probabilities.

edge subtending zero angle, *e.g.*, the one marked as g_0 in Fig. 5*a*. It is a distinct possibility that the target may exit g_0. We must relax our initial assumption and incorporate this situation. To do this, we expand every gap edge by a predefined distance δ and call the resulting region the *gap zone*:

$$\mathcal{G}(g) = \{x \in \mathcal{V} \mid d(x, g) \leq \delta\},$$

where $d(x, g)$ denotes the shortest distance from x to g. Now the heading probability of g depends on the angle subtended by its gap zone instead of itself. In general, adjacent gap zones may overlap, and the probability in overlapping region must be split evenly among all gap zones involved. Taking all these into account, we have the following formula for computing the heading probability of g:

$$p_g = P(G) = \int_{\Theta_G} f(\theta)/h(\theta)\, d\theta, \tag{12}$$

where Θ_G is the angular range subtended by the gap zone G of g and $h(\theta)$ is the number of gap zones that λ_θ intersects. Note that $h(\theta) \geq 1$.

The threshold δ for determining the gap zone basically says that the target may exit g whenever it comes within a distance δ of g. It can be chosen according to our understanding of target behaviors. In our experiments, we chose δ to be the distance that the target can reach with maximum velocity. This is an aggressive choice, indicating high confidence in the target motion model.

Good velocity prediction helps the robot to focus on the important gap edges and improve tracking performance. Consider the example in Fig. 6. It compares our new tracking strategy with the one in [5], [9], which does not use velocity prediction. Each image in Fig. 6 shows several small line segments rooted at the current robot position. Each segment corresponds to the heading probability of a gap edge. The length of the segment is proportional to the heading probability, and its orientation points to the gap edge associated with the heading probability. For the old strategy, all the gap edges are weighted with equal probabilities. For the new strategy, the gap edge to which the target is headed has a distinctively large heading probability, indicated by a long segment.

When the target makes abrupt turns, the velocity prediction is usually inaccurate. Our tracking strategy may cause the robot to make the wrong move. Consider the example in Fig. 7. The target makes several abrupt turns. However, the velocity

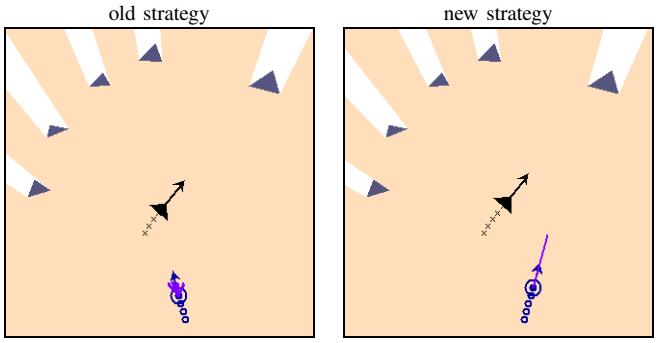


Fig. 6. Using the estimated target velocity information, the robot can focus on the important gap edges.

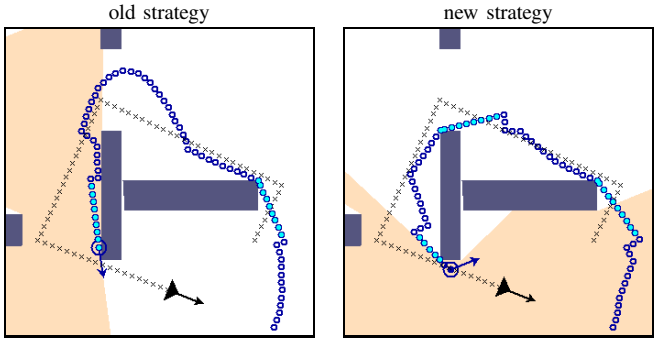


Fig. 7. An example in which the target makes abrupt turns.

prediction is reasonable for most of the time. Despite the wrong moves, our tracking strategy follows the target to the end and performs better than the old strategy, which loses the target midway. The advantage of velocity prediction is further confirmed by the experiments presented in Section V.

D. Emergency Actions

Our tracking strategy invokes two emergency actions, when the target is dangerously close to escape. We describe below the specific robot actions when the target is in region I of a gap edge g. The actions for the other cases are simpler. We omit the details for lack of space.

First, we estimate the escape time t_{esc} based on e and the current velocities of the robot and the target. If t_{esc} is below a threshold, then the robot must reduce the immediate risk maximally by increasing its tangential motion w.r.t. g. It does this by setting $v = V\hat{\mathbf{t}}$.

Next, if the target indeed escapes from an occlusion edge g, the best that the robot can hope for is to eliminate g. A naive way is to run directly towards the corresponding occlusion point by setting $v = V\hat{\mathbf{r}}$. The fastest way of eliminating g requires knowledge of an obstacle edge lying outside the robot's visibility set. In this case, the optimal robot actions is to swing out along a suitably constructed circular path [6].

V. Experiments in Simulation

We implemented our new tracking strategy in C++ and compared it with earlier work [5] in simulation. To have a fair comparison, we provided the old strategy the same emergency actions that our new strategy uses, though they are not in the

TABLE I

PERFORMANCE COMPARISON OF THE OLD AND THE NEW TRACKING STRATEGIES.

Env.	Total No. Target Steps	Old Strategy		New Strategy	
		No. Steps Visible (%)	No. Times Lost (Steps Lost)	No. Steps Visible (%)	No. Times Lost (Steps Lost)
Maze	82	35 (43%)	2 (11,12)	74 (90%)	1 (8)
City Blocks	156	78 (49%)	6 (14, 15, 16, 15, 8, 10)	131 (84%)	2 (13, 12)

original work. Some of the comparison results have already been presented in Figs. 3, 4, 6, and 7. Here, we show two additional examples with more complex geometry (Fig. 8):

a) Maze. This environment brings together various geometric features, such as long corridors, open spaces, and sharp turns. The target takes a long and winding path. Even with emergency actions, the old strategy loses the target midway. The new strategy follows the target to the end. It loses the target once, but recovers it quickly through emergency actions.

b) City blocks. This example mimics city blocks in an urban environment. The old strategy has lots of difficulty in this environment. It loses the target many times for extended periods (see Table I) and fails to follow the target to the end. The new strategy has much improved performance.

Detailed performance statistics on these two environments are shown in Table I. Column 2 of the table lists the length of the target trajectory in time steps. For the old strategy, column 3 lists the number of steps that the robot has the target visible as well as the number as a percentage of the total number of target steps. Column 4 lists the number of times that the target is lost *and* recovered with emergency actions, as well as the durations for which the target is lost. Columns 5–6 give the same information for our new strategy. The comparison in these two environments shows that the new strategy (i) less likely loses the target, (ii) has the target visible for much longer total duration, and (iii) always follows the target the end. All these indicate better performance.

VI. CONCLUSION

This paper presents a practical algorithm for target tracking, an interesting class of motion planning problems that combine the usual motion constraints with robot sensors' visibility constraints. We introduced the notion of vantage time, which provides a systematic way to integrate various factors contributing to the escape risk. It is used to formulate a risk function and construct a greedy tracking strategy. We compared our new strategy with earlier work in extensive simulation experiments and obtained much improved results. We believe that the improvements result from a better-formulated risk function that takes into account both the relative positions of the robot and the target and the velocity of the target.

We plan to implement our algorithm and test it on real robots. For this, we will improve the robot motion model, incorporating nonholonomic constraints for wheeled robots if necessary. We will also improve the robot sensor model by imposing limited viewing angles. Another interesting extension is to use this approach for multi-robot tracking [12].

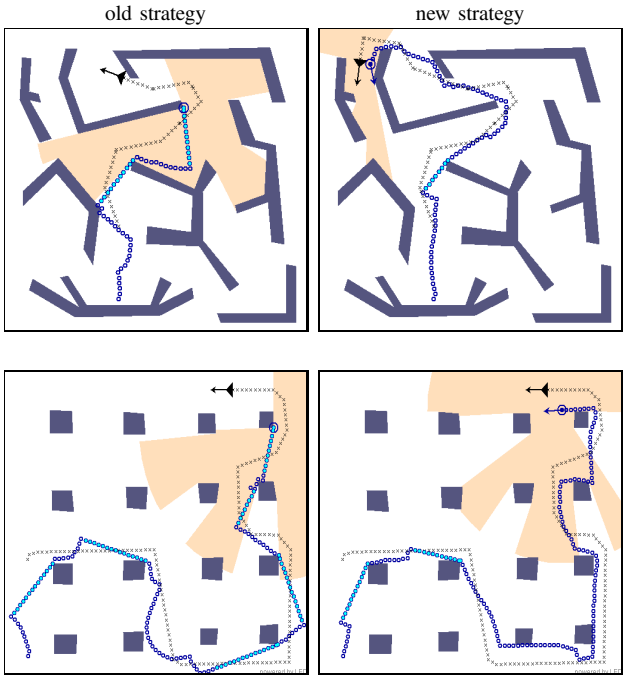


Fig. 8. Two environments with complex geometry.

REFERENCES

[1] T. Bandyopadhyay, Y. Li, M. Ang Jr., and D. Hsu, "Stealth tracking of an unpredictable target among obstacles," in *Algorithmic Foundations of Robotics VI*, M. Erdmann *et al.*, Eds. Springer-Verlag, 2004, pp. 43–58.

[2] E. Birgersson, A. Howard, and G. Sukhatme, "Towards stealthy behaviors," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*, 2003, pp. 1703–1708.

[3] A. Efrat, H. González-Baños, S. Kobourov, and L. Palaniappan, "Optimal strategies to track and capture a predictable target," in *Proc. IEEE. Int. Conf. on Robotics & Automation*, 2003, pp. 3789–3796.

[4] P. Fabiani, H. González-Baños, J. Latombe, and D. Lin, "Tracking a partially predictable target with uncertainties and visibility constraints," *J. Robotics & Autonomous Systems*, vol. 38, no. 1, pp. 31–48, 2002.

[5] H. González-Baños, C.-Y. Lee, and J.-C. Latombe, "Real-time combinatorial tracking of a target moving unpredictably among obstacles." in *Proc. IEEE Int. Conf. on Robotics & Automation*, 2002, pp. 1683–1690.

[6] V. Isler, S. Kannan, and K. Daniilidis, "Local exploration: Online algorithms and a probabilistic framework," in *Proc. IEEE. Int. Conf. on Robotics & Automation*, 2003, pp. 1913–1920.

[7] J. Latombe, *Robot Motion Planning*. Boston, MA: Kluwer Academic Publishers, 1991.

[8] S. LaValle, H. González-Baños, C. Becker, and J. Latombe, "Motion strategies for maintaining visibility of a moving target," in *Proc. IEEE Int. Conf. on Robotics & Automation*, 1997, pp. 731–736.

[9] C.-Y. Lee, "Real-time target tracking in an indoor envrionment," Ph.D. dissertation, Dept. of Aeronautics & Astronautics, Stanford University, Stanford, CA, USA, 2002.

[10] M. Marzouqi and R. Jarvis, "Covert robotics: Covert path planning in unknown environments," *Proc. Australian Conf. on Robotics & Automation*, 2003.

[11] R. Murrieta, A. Sarmiento, and S. Hutchinson, "A motion planning strategy to maintain visibility of a moving target at a fixed distance in a polygon," in *IEEE Int. Conf. on Robotics & Automation*, 2003.

[12] R. Murrieta-Cid, H. H. González-Baños, and B. Tovar, "A reactive motion planner to maintain visibility of unpredictable targets." in *Proc. IEEE. Int. Conf. on Robotics & Automation*, 2002, pp. 4242–4248.