Real-Time Tracking of an Unpredictable Target Amidst Unknown Obstacles

Cheng-Yu Lee(1) **Hector González-Baños**(2) **Jean-Claude Latombe**(1)

(1) Robotics Laboratory, Computer Science Department, Stanford University, CA, USA
(2) Honda's Fundamental Research Labs., Mountain View, CA, USA
Email: chengyu@robotics.stanford.edu, hhg@hra.com, latombe@cs.stanford.edu

Abstract

This paper describes a new approach to keep a moving target into the field of view of a robot, despite obstacles. Unlike previous work, this approach does not assume that a complete or partial model of the target's behavior is available, nor that a map of the obstacles is given. At approximately 10 Hz, the robot constructs a local map of the environment from range data, localizes the target in this map, and computes its new velocity command. This command is selected to minimize a function that measures the risk that the target may escape the robot's field of view by crossing an occlusion ray created by an obstacle. The risk function defined in this paper combines a reactive term, whose minimization handles immediate threats, and a look-ahead term, whose minimization improves the robot's future ability to react to evading motions of the target. The experimental robotic system has been remarkably successful at tracking a target performing brisk maneuvers and swift turns behind obstacles.

1 Introduction

This paper describes a robot, called the *observer*, whose task is to keep a moving target into its field of view. While previous tracking techniques assume that both a model of the target's bahavior and a map of the obstacles are given, our approach makes no such assumptions. At approximately 10 Hz, the implemented observer constructs a local map of the environment from sensed range data, localizes the target in this map, and computes a new velocity command that is sent to its servo-controller. This command is selected to minimize a measure of the risk that the target gets hidden by obstacles.

Our observer is a mobile robot (SuperScout platform of Nomadic Technology) equipped with a polar laser range sensor (see Fig. 1). The target is another mobile robot. The observer and the target operate in an in-door environment with solid and mostly static obstacles. The observer's range finder provides a 180-dg horizontal scan of the environment. Both the target's position and the local obstacle map are extracted from this scan. Despite the paucity of its sensing equipment, our observer has been remarkably successful at tracking a target performing brisk maneuvers and swift turns behind obstacles in non-engineered environments. The observer also tolerates transcient obstacles, though it does not explicitly distinguish them from stationary obstacles. We attribute much of its success to both our new risk-based tracking approach and the specific risk function embedded in this approach, which combines a reactive and a look-ahead term. These concepts are presented in this paper.

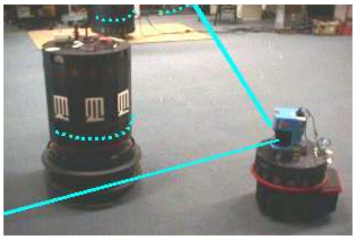


Figure 1: The observer (right) and the target

2 Previous Work

Automatic target tracking and interception. Tracking has been widely studied in the field of automatic control, for such applications as missile control [13]. The tracker and the target moves in 3-D space. Unlike in our work, obstacles are few (if any) and occlusion of the target is not a concern. Instead, the emphasis is on dealing with tight dynamic restrictions on the tracker's motion.

Visual tracking. Here the goal is to track an object in a sequence of images [7], despite partial occlusions of the target by other objects, and aspect changes and self-occlusions of the target as it moves. There is no attempt to prevent occlusions from happening by moving the camera appropriately. Visual tracking and sensor placement are two complementary subproblems in target tracking among obstacles. Our work addresses the second subproblem.

Visual servo-control. Here visual tracking is used to control the motion of a robot [6]. The robot has a certain goal, like docking against another machine. Visual feedback is used to track an object – e.g., the machine against which the robot intends to dock – in order to adjust the robot's trajectory. Again, there is no attempt to move the camera to avoid undesirable occlusions.

Guarding an art gallery. The basic art-gallery problem is to compute the positions of fixed observers (guards), such that they collectively see a given space [12]. Finding the minimum number of guards in a polygonal space is NP-hard, but approximation algorithms exist [4]. Such an algorithm could be used to compute the placements of sensors such that the target would always be visible from one of them. Instead, our work assumes that there is only one sensor, but this sensor moves.

Off-line planning. A similar problem as ours is considered in [2], assumming that both a global map of the obstacles and the target's trajectory are given in advance. An off-line planner backprojects visibility regions to compute a tracking trajectory. This trajectory is then executed in open loop by the observer. The assumptions are too constraining to lead a practical robotic system, but this method has been applied to computer graphic animation [10].

Game-theoretic approach. The tracking algorithm in [8] is based on a game-theoretic approach and minimizes a cost function using dynamic programming. When the target's trajectory is known in advance, along with the global obstacle map, an optimal trajectory is computed off-line. When the target's trajectory is not known, a probabilistic model of the target's behavior is used to compute and update in real-time a strategy maximizing the probability that the observer keeps the target in view within some time horizon h. But the computational cost grows quickly with h, hence h must be small. Systems built using this method [1, 8, 11] give reasonably good results when the target does not move too swiftly. The risk-based method presented below does not require a model of the target's behavior nor does it use costly dynamic programming. Experimental results show that it is faster and handles swift targets more reliably.

Observer self-localization. When the observer uses a global map, it must localize itself in this map. The above game-theoretic approach is extended in [3] to take into account the impact of self-localization uncertainties on tracking. A cost function allows the observer to decide at each stage of the tracking process whether it is preferable to perform the best tracking motion, or to deviate from this motion in order to see a landmark and achieve better self-localization. Our method does not make use of a prior global map. It eliminates the need for the observer to localize itself, hence saving precious computational time and avoiding sporadic major self-localization errors.

3 Tracking Procedure

Our tracking approach is based on minimizing a function φ of the observer's position that estimates the *risk* that the target may escape the observer's field of view by crossing an occlusion ray created by an obstacle. The commanded velocity is set to $-\nabla\varphi$. In our implementation it is updated at 10 Hz.

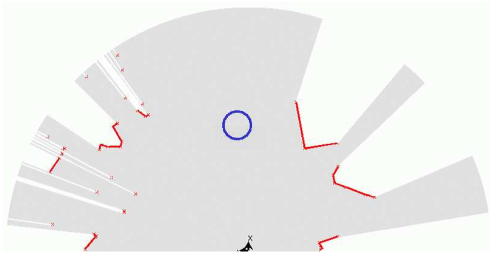


Figure 2: Range scan and extracted information

Each sensor reading provides a 180-dg horizontal scan of the environment made of 360 points at regular angular spacing. This scan is first segmented into maximal sequences of adjacent points. The target is then recognized (as a circular arc of known radius) and localized in the observer's coordinate system. Polygonal lines (polylines) are fit through the remaining sequences of points. Each endpoint of a polyline is an *occluding vertex* through which an *occluding ray* is erected. Fig. 2 shows the outcome of the treatment of a scan. The localized target is shown in plain (blue) line. The occlusion rays are the straight line segments converging toward the observer (the black point at the base of the figure). The outer circular arcs bounding the shaded area corresponds to the sensor's maximum range.

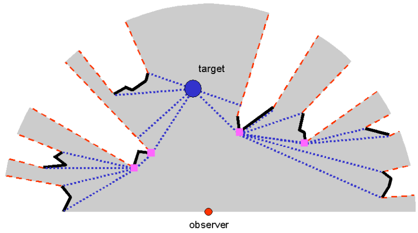


Figure 3: Escape-path tree

The tracking algorithm then computes the shortest escape path of the target to every occlusion ray. The obtained paths form a tree that we call the *escape-path tree* (EPT for short). An EPT is shown in Fig. 3 (here, the paths have been computed as if the target was a point). The paths in the EPT are used to assign an individual escape risk φ_e to each occlusion ray e. These individual risks are aggregated into a global risk function φ. As indicated below, this function is differentiable, hence the algorithm easily calculates $-\nabla\varphi$.

The overall tracking algorithm is the following:

Procedure TRACK

Repeat:

1. Acquire a scan.

2. Localize the target.

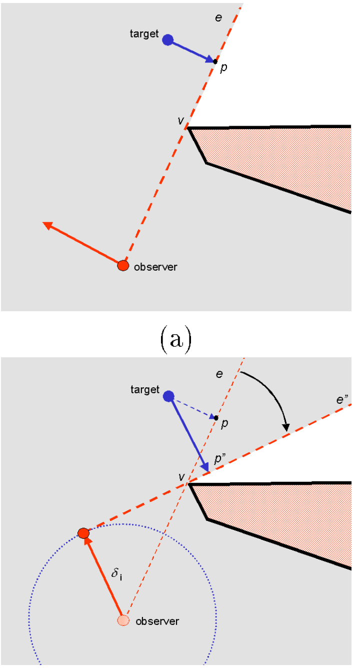
3. If step 2 failed, then exit with failure.

4. Extract occluding vertices and occlusion rays.

5. Compute the EPT.

6. Set the velocity command to $-\nabla\varphi$.

An efficient algorithm given in [5, 9] computes the EPT by exploiting geometric properties of shortest escape paths in star-shaped polygons. Below, we focus on the description of the risk function φ.



(a)

(b)

Figure 4: Approximate strategies

4 Risk Function

In the absence of any model of the target's behavior, a tracking approach suggested in [8] is to compute the velocity command that maximizes the escape time for the worst-case trajectory of the target. However, when a global map of the obstacles is given, this computation is extremely costly. When no map is given, it is simply impossible. This led us to investigate approximations of this strategy.

4.1 Escape Risk for Occlusion Ray

For a moment, let us focus our attention to a single occlusion ray e extracted from the current scan, and let τ_e denote the shortest escape path of the target to e. If the target's motion is not restricted by any nonholonomic or dynamic constraint, the ratio of the length of τ_e by the target's maximal velocity is exactly the minimum time ε that the target needs to escape the *current* field of view of the observer through e. If the target's motion is restricted, this ratio remains in general a good lower-bound approximation of this time. Hence, a possible approximate strategy for the observer is to move in the direction that yields the greatest differential increase of the length of τ_e. To illustrate consider the example of Fig. 4(a), where the path τ_e consists of a single line segment between the target's position and an escape point p in an occlusion ray e erected from the occluding vertex v. The observer's motion that yields the greatest differential increase of the length of τ_e is perpendicular to the line supporting e.

We can push this reasoning further. During the time ε it would take for the target to reach p by moving at maximal velocity along τ_e, the observer can reach any point within a circle of radius δ centered at its current position, where δ is the product of the observer's maximal velocity by ε (assuming that no nonholonomic or dynamic constraint limits the observer's motion). Consider the tangent line to this circle drawn from v. A better direction of motion for the observer than the vector shown in Fig. 4(a) is the vector shown in Fig. 4(b), which points toward the tangency point. It combines a component perpendicular to the line supporting e and a component in this line pointing toward v. The key observation that must be made here is that the time needed by the target to escape the observer's current field of view provides some "slack" that the observer can exploit to reach a position where it will have greater ability to react to future evading moves of the target. More specifically, by getting closer to the occluding vertex v, while it still has time to do so, the observer will later be able to perform swinging motions leading the occluding edge e to rotate at a faster rate around v.

This discussion yields the risk function φ_e for one occlusion edge e. We define φ_e to be of the form:

$$\varphi_e = c\,r^2\,f(1/h), \qquad (1)$$

where c is a scaling factor, r is the distance between the observer's position and the occluding vertex v, h is the length of the shortest path τ_e ending in e, and f is an increasing function. The experimental results of Section 5 where obtained for $f(x) = \ln(x^2+1)$. Hence, φ_e goes to $+\infty$ when h goes to 0, and to 0 when the observer gets arbitrarily close to v.

One can regard φ_e as being made of two terms:

1. The *reactive* term, $f(1/h)$: Its minimization in-

creases the length of the escape path to e.

2. The *look-ahead* term, r^2: Its minimization improves the future ability of the observer to react to target's evading moves.

The component of the commanded velocity associated with φ_e is:

$$-\nabla\varphi_e \ = \ -2crf(1/h)\nabla r + c\frac{r^2}{h^2}f'(1/h)\nabla h.$$

It can easily be computed in closed form by treating r and h as functions of the observer's position [5, 9]. When h is very small, there is an immediate threat; then, the dominant term is the reactive term and the commanded velocity is almost perpendicular to the supporting line of e (direction of ∇h). Instead, if h is large, the dominant term is the look-ahead component and the commanded velocity has a substantial component pointing toward v (direction of $-\nabla r$).

4.2 Global Risk Function

Let us now consider all occlusion rays and all escape paths in the EPT. We use Eq. (1) to compute the risk associated with every occlusion edge. Note that the shortest escape path (over all possible occlusion edges) may not result in the greatest risk. An occlusion ray with a longer escape path to it may represent a greater risk if the observer is located further away from this ray's occluding vertex. So, the issue is: How can we combine the individual risks associated with the occlusion edges to define the global risk φ?

One possibility is to identify the occlusion ray e that yields the greatest risk φ_e and set $\varphi = \varphi_e$. But, as the observer and the target move, the greatest-risk occlusion ray, and so the commanded velocity, may change abruptly and frequently. In an early implementation, this resulted into a shattering control causing erratic moves of the observer. To avoid this problem, we define the global risk function φ by averaging the escape risks associated with several occlusion rays.

In the "standard" average, the risks associated with all occlusion rays are counted equally. But many escape paths often share the same initial structure. Such paths are then over-represented in φ at the expense of the solitary escape paths. Two averages that work better in practice are the following:

- *Recursive average*: The risks of the EPT nodes sharing the same parent are first averaged and the result is propagated upward to the parent.

- *Tournament average*: The global risk is the average of the risks associated with the k greatest-risk occlusion rays (with $k = 3$ or 4).

Since the gradient is a linear operation, we apply directly these averaging techniques to the gradients of the individual risks. No risk φ_e, nor φ, is ever computed by the tracking algorithm. The experimental results presented below were obtained with the recursive average.

5 Experimental Results

We have implemented the above tracking approach a SuperScout mobile robot of Nomadic Technologies equipped with an LMS200 laser range finder of Sick OpticElectronic. The tracking algorithm runs on an off-board 410MHz Pentium II processor connected to the robot by wireless ethernet. The target, a joysticked Nomad-200 robot, is modeled as a disc of non-zero radius and the escape paths are computed accordingly. The SuperScout robot is an almost cylindrical nonholonomic vehicle with zero-turning radius. A simple trajectory follower is used to track the flow of the successive velocity commands $-\nabla\varphi$ [5, 9]. In addition, we use a second regulator to deal with the fact that the range sensor has limited range (8 meters) and a 180-dg cone. This regulator is a simple proportional feedback of the target's deviation from an ideal position in the sensor's field of view. The maximal linear velocities of the observer and the target are approximately the same, with the observer slightly faster (0.5 m/sec).

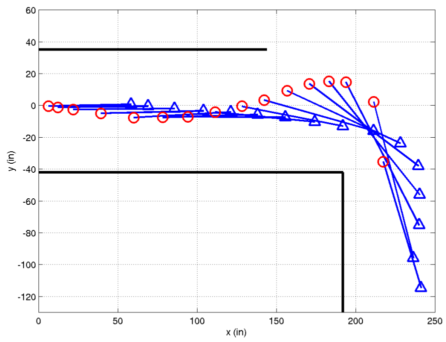


Figure 5: Tracking a target around a corner

We have conducted various experiments with this system. In some experiments, we have compared the risk-based algorithm with a pure visual-tracking algorithm. This second algorithm only tries to keep the target close to an ideal position in the sensor's field of view. It ignores obstacles, hence does not take potential occlusion into account. Unlike the risk-based algorithm, the pure visual-tracking algorithm is easily tangled by a target moving toward an obstacle corner and swiftly turning around this corner. Fig. 5 shows a record of the target's and observer's trajectories in one experiment with the risk-based algorithm. The successive positions of the target and the observer are shown with triangles and circles, respectively. Segments connects the pairs of simultaneous positions. Note how well the observer anticipates the risk that the target may turn around the corner and swings around this corner.

We have also successfully experimented with our observer in environments with transient obstacles, obtained by suddenly adding an obstacle in the field of view of the observer and either leaving it there, or removing it a short amount of time later. The observer has proven to be tolerant to such obstacles [9].

Fig. 6 shows snapshots of another experiment where the observer successfully tracks the target along a long tour throughout our laboratory environment. The tour starts in a corridor, then traverses a large area with many different objects (chairs, boxes), and finally ends in another corridor. The scans and EPT's corresponding to these snapshots are displayed in Fig. 7.

In a number of experiments, the observer also failed to track the target correctly. Most failures occurred because the observer either failed to detect the target though it was visible, or mis-located it. Such failures occurred mainly in the presence of cylindrical objects (e.g., waste baskets) similar to the target. But imagine yourself having to track a target in a dark environment by using a flashlight projecting only a plane of light! Given the paucity of its sensor, our observer has been remarkably successful.

6 Conclusion

This paper describes a new approach to target tracking based on risk minimization. It proposes a risk function that combines a reactive term, to deal with immediate threats, and a look-ahead term, to better use available time. Experiments have shown that the observer is able to keep the target in sight in both densely and sparsely cluttered environments, even when the target performs sudden and swift moves. An obvious improvement is to equip the observer with an additional sensor, for instance a stereo-vision system, to better detect and localize the target. An interesting extension will be to perform tracking and environment mapping simultaneously. A more difficult extension would be to coordinate a team of independent observers with limited communication to track multiple targets [11].

Acknowledgements: We thank Vicky Chao for her help in running the experiments.

References

[1] H. González-Baños, J.L Gordillo, J.C. Latombe, D. Lin, A. Sarmiento, C. Tomasi, "The Autonomous Observer: A Tool for Remote Experimentation in Robotics," *SPIE Proc. Telemanipulator and Telepresence Technologies VI*, M. Stein (ed.), vol. 3840, pp. 210-221, 1999.

[2] C. Becker, H. González-Baños, J.C. Latombe, C. Tomasi, "An Intelligent Observer," *Proc. 4th Int. Symp. on Experimental Robotics (ISER)*, pp. 153-160, 1995.

[3] P. Fabiani, H. González-Baños, J.C. Latombe, D. Lin, "Tracking a Partially Predictable Target with Uncertainties and Visibility Constraints," *J. of Robotics and Autonomous Systems*, 38(1):31-48, 2002.

[4] H. González-Baños, J.C. Latombe, "A Randomized Art-Gallery Algorithm for Sensor Placement," *Proc ACM Symp. on Computational Geometry*, pp. 232-240, 2001.

[5] H. González-Baños, C.Y. Lee, J.C. Latombe, "Real-Time Combinatorial Tracking of a Target Moving Unpredictably Among Obstacles," *Proc. Int. Conf. Robotics & Automation*, 2002.

[6] S. Hutchinson, G.D. Hager, P.I. Corke, "A Tutorial on Visual Servo Control," *IEEE Tr. Robotics & Automation*, 12(5):651–670, 1996.

[7] D.P. Huttenlocher, J.J. Noh, W.J. Rucklidge, "Tracking Non-Rigid Objects in Complex Scenes," *Proc. 4th Int. Conf. on Computer Vision*, pp. 93–101, 1993.

[8] S. LaValle, H. González-Baños, C. Becker, J.C. Latombe, "Motion Strategies for Maintaining Visibility with a Moving Target," *Proc. IEEE Int. Conf. Robotics & Automation*, pp. 731–736, 1997.

[9] C.Y. Lee, *Reat-Time Target Tracking in an Indoor Environment*, Ph.D. Thesis, Dept. of Aeronautics and Astronautics, Stanford University, Stanford, CA, June 2002.

[10] T.Y. Li, T.H. Yu, "Planning Tracking Motions for an Intelligent Virtual Camera," *Proc. IEEE Int. Conf. Robotics & Automation*, pp. 1353–1358, 1999.

[11] R. Murrieta-Cid, H. González-Baños, B. Tovar, "A Reactive Motion Planner to Maintain Visibility of Unpredictable Targets," *Proc. IEEE Int. Conf. Robotics & Automation*, 2002.

[12] J. O'Rourke, "Visibility," in *Handbook of Discrete and Computational Geometry*, J.E. Goodman and J. O'Rourke (eds.), CRC Press, Boca Raton, FL, pp. 467-479, 1997.

[13] T.L. Song, T.Y Um, "Practical Guidance for Homing Missiles with Bearings-Only Measurements," *IEEE Tr. Aerospace and Electronic Syst.*, 32:434–443, 1996.

Figure 6: Tracking a target performing a long tour

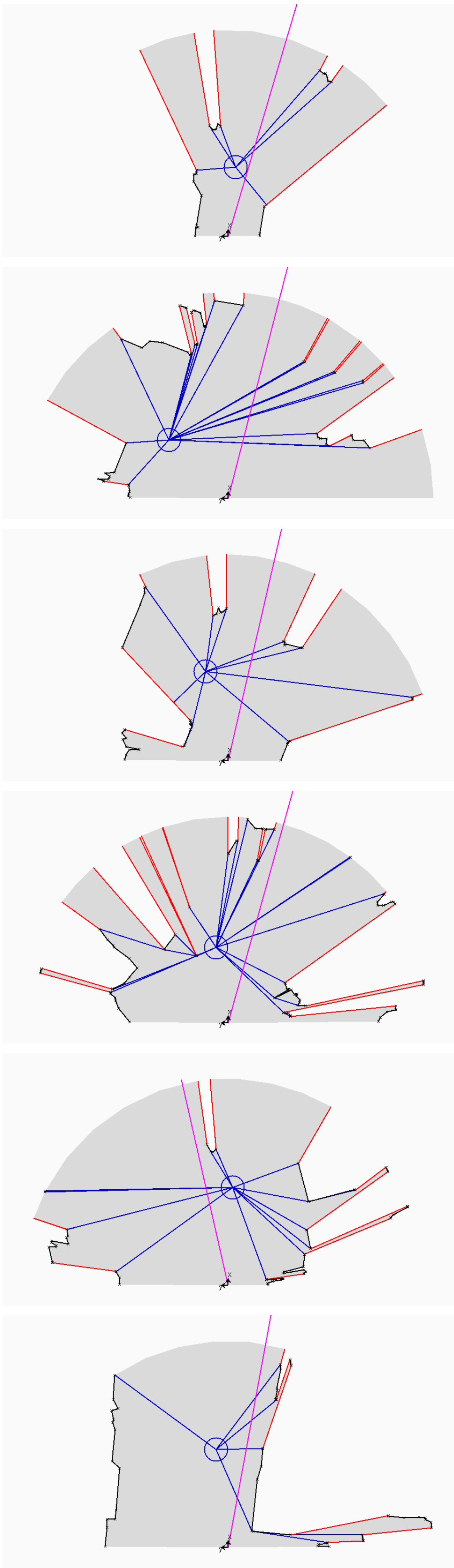


Figure 7: Scans and EPT's