

# Following Multiple Unpredictable Coherent Targets Among Obstacles

Christopher Vo

Jyh-Ming Lien \*

## Abstract

Camera control is essential in both virtual and real-world environments. The quality of the camera’s placement and motion may spell the difference between usability and confusion. Our work focuses on an instance of camera control called *target following*, and offers an algorithm for following multiple targets with *unpredictable* trajectories, among known obstacles. To the best of our knowledge, this work is the first attempting to address this important problem.

## 1 Our Method and Results

In *multi-targets following*, the camera’s primary objective is to follow and maximize visibility of multiple moving targets. For example, in video games, a third-person view camera may be controlled to follow a group of characters through complicated virtual environments. In robotics, a camera attached to robotic manipulators could also be controlled to observe live performers in a concert, monitor assembly of a mechanical system, or maintain task visibility during teleoperated surgical procedures. In general, it is difficult for a user to manually control the camera while also concentrating on other critical tasks. Therefore, it is desirable to have an *autonomous camera system* that handles the camera movement. The camera control problem has been studied extensively in both computer graphics and robotics (see a recent survey [Christie et al. 2008]) because of its broad applications. Unfortunately, many of these methods are not applicable to real-time multi-targets following *among obstacles*.

The problem of building a camera for following the targets can be viewed as an online motion planning problem, in which the planner must be able to generate a trajectory for the camera and predict targets’ motions in real time. In this poster, we present a *sampling-based planner* to robustly follow multiple targets among obstacles. We assume that the workspace is populated with known obstacles represented by polygons. These polygons are the projection of 3D objects that can potentially block the camera’s view. This projection essential reduces our problem to a 2D workspace. We assume that, initially, the targets  $T$  are visible by the camera  $C$ , and, during the entire simulation,  $T$  exhibit certain degree of coherence (similar to a flock of bird). The targets are either controlled by the user or by another program, so the trajectories of the targets are not known in advance. However we assume that the maximum (linear) velocity of the target,  $v_T^{max}$ , is known. The current positions  $X_{T'}(t)$  of some targets  $T' \subset T$  at time  $t$  are known only if  $T'$  is in  $C$ ’s view range. The camera  $C$  also has bounded linear velocity  $v_C^{max}$ . The camera  $C$ ’s view range  $\mathcal{V}_C$  is defined as a tuple:  $\mathcal{V}_C = (\theta, r_{near}, r_{far})$ , where  $\theta$  is the view angle, and  $r_{near}$  and  $r_{far}$  define the near and far view range. The exact configuration of this view range at time  $t$ , denoted as  $\mathcal{V}_C(t)$ , is defined by the tuple and the camera’s location  $x_C(t)$ . The position  $x_C$  of the camera is simply governed by the following linear equation:  $x_C(t + \Delta t) = x_C(t) + \Delta t \cdot v_C(t)$ . The problem of target following then can be defined as: finding a sequence of velocities  $v_C(t) \leq v_C^{max}$ , so that  $x_C(t), \forall t$ , is collision free, and  $\sum_t \text{card}(\{T' \subset T \mid X_{T'}(t) \subset \mathcal{V}_C(t)\})$  is maximized.

Our approach is a sampling-based method extended from [Becker et al. 1997], which is originally designed to track a single target.

\*Authors are with Department of Computer Science, George Mason University, Fairfax, VA, USA, {cvo1, jmlien}@cs.gmu.edu

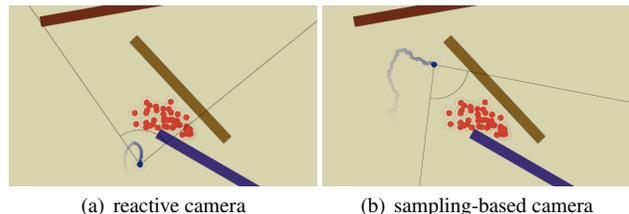


Figure 1: Difference between the reactive and proposed cameras.

At each time step, given the visible targets  $T$ , the planner first creates  $k$  point sets  $P_T$ , where  $k$  is a parameter. Each point set contains  $|T|$  predicted target positions. The predicted position of each target  $\tau$  is sampled from the region visible from  $\tau$ , and is at most  $(v_T^{max} \cdot \Delta t)$  away from  $\tau$ . The planner then creates a set  $P_C$  of camera configurations that are at most  $(v_C^{max} \cdot \Delta t)$  away from  $C$ . To decide the next camera configuration, we simply determine  $\arg \max_{x \in P_C} (\sum_{X \in P_T} \text{vis}(x, X))$ , where  $\text{vis}(x, X)$  is the number of points in  $X$  that are visible by  $x$ .

We also developed a reactive camera for comparison. This reactive camera determines its next configuration by maximizing the number of visible targets based only on the targets’ currently positions. Figure 1 shows the difference between the reactive and the proposed cameras situated at the same target configuration. The latter clearly maintains better visibility when the targets move through the narrow corridor. Figure 2 shows a stronger evidence that the proposed method can follow the targets better than the reactive camera. A main limitation of the sampling-based planner is the smoothness of the planned camera path. However, smoothness can usually be improved via a post-processing step.

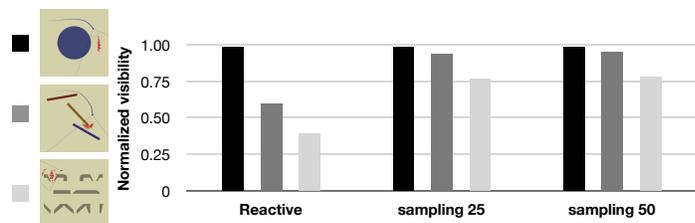


Figure 2: Following 50 targets in three environments. Quality is measured by the normalized visibility of the reactive camera vs. two sampling-based cameras, named “sampling 25” and “sampling 50”, which sample 25 and 50 camera configurations, respectively, in every time step. Normalized visibility is the ratio of the visible targets during the entire simulation. Each bar is an average over 30 runs, each of which is a 9000 time-step simulation.

## References

- BECKER, C., GONZÁLEZ-BANOS, H., LATOMBE, J.-C., AND TOMASI, C. 1997. An intelligent observer. In *The 4th International Symposium on Experimental Robotics IV*, Springer-Verlag, London, UK, 153–160.
- CHRISTIE, M., OLIVIER, P., AND NORMAND, J.-M. 2008. Camera control in computer graphics. *Computer Graphics Forum* 27, 8, 2197 – 2218.