Single Camera Multiplexing for Multi-Target Tracking

Chiraz BenAbdelkader, Philippe Burlina, and Larry Davis
Computer Vision Laboratory
University of Maryland
College Park, MD 20742 USA
chiraz,burlina,lsd@cfar.umd.edu

Abstract

This paper considers the problem of designing a real-time surveillance system, equipped with a single camera on a pan/tilt platform, to track multiple moving targets within the camera's field of regard. The objective is to maintain motion trajectory information of as many of these targets as possible, and for as long as possible. Because the camera can only capture a fraction of the field of regard at any one time, it may not be possible to track all targets of interest at once. The problem can hence be viewed as one of time-sharing a scarce resource (the camera) among multiple contending users (the targets). For this, we propose a system architecture that consists of two modules: one that handles the high-level time-allocation (target scheduling) issue within a queuing theory framework, and another that handles low-level localized detection and tracking of a target, based on principles of recursive filtering (estimation) and feedback control. Effective operation of the system relies on ability to accurately determine the two key parameters for each target: how much tracking time to allocate to it, and how often to re-schedule it.

1. Introduction

Recent years have seen a continued increase in the need for and use of automatic video surveillance, both in urban (civilian) and military airborne applications. A surveillance system is typically comprised of one or more video sensors (such as cameras), each of which is mounted on a mobile pan/tilt platform. Because the platform hardware is expensive, making the most use out of each sensor is a worthy goal. To this end, we consider the design of a novel real-time surveillance system that uses a single camera equipped with pan/tilt and zoom capabilities, to 'keep track' of as many targets as possible for as long as possible.

To be able to accommodate multiple targets scattered anywhere within the camera's field of regard (and that may not necessarily be captured within one field of view), the camera will need to be time-shared among multiple fields of view, and will stay on any one FOV for only a limited amount of time. Two main issues arise. The first is, how and whether it is possible to maintain the trajectory of a target without continuously tracking it. The second is, how to manage the time-allocation of the camera among these contending targets. Accordingly, the system architecture we describe in this paper consists of two modules that roughly handle each one of these issues. Specifically, the two modules operate in tandem, wherein the planning module selects one target at a time to be tracked for a prescribed amount of time, and the control/tracking module transparently tracks the designated target within the allotted amount of time.

We shall model the planning module as a queuing system that schedules access to a scarce resource (the camera/platform) among multiple contending users (the targets). Two key parameters need to be specified for each user in this model: its *service time* and its *think time*, which respectively correspond to how much time the control/tracking module may spend on any one target, and how soon after it was tracked a target will need to be re-acquired for tracking. These two parameters are not known a priori, but need to be determined by the system. We shall propose heuristic methods for doing this based on the error covariance of the motion trajectory estimate provided by the control/tracking module.

The control/tracking module, on the other hand, will be modeled as a recursive stochastic filter to estimate the target's motion based on noisy measurements of its position in the image, and a negative feedback loop to control the orientation of the camera and make sure the target stays within the camera's field of view during tracking.

2. Assumptions

- The platform is stationary, and is equipped with a pan/tilt mechanism that rotate the camera. The range of the pan and tilt rotation angles, together with the position of the camera with respect to the targets' surface of motion, determine the camera's field of regard. The internal calibration parameters and the position of the camera are assumed to be known at all times.

- An external mechanism, such as a Moving Target Indicator (MTI), does the initial detection of moving targets in the camera's field of regard and cues our system with their initial 3D positions. The issue of the latent time between when the target is detected and when it is first scheduled for tracking will be dealt with appropriately in the planning module.

- The targets are moving on a known surface and their motion are well- behaved so that their motion is sufficiently modeled with first-order dynamics.

3. Control/Tracking Module

The goal of this module is to track a given moving target for a prescribed amount of time, and return the estimated motion trajectory, which obviously requires adaptive control of the camera to ensure that the target remains visible. The module consists of three main computational sub-modules, that operate in a feedback loop as follows.

- The *target detection* module provides the low-level image operations needed to detect the target of interest, and determine its location in the current image frame. The issues involved in this task have been addressed at length in previous computer vision literature, such as [6] [3] [4], and so we will not elaborate on it.

- The *data filtering* module estimates the target's 3D motion trajectory based on noisy image measurements provided by the detection module. Here we use a first-order dynamic model for target motion, and the Extended Kalman Filter (EKF) algorithm to recursively estimate the parameters of this model (i.e. position and velocity) [1] [5] [7].

- The *camera control* module decides when and how to rotate the camera, based on the estimated 3D position of the target and the current position of the camera, and taking into account the delay incurred by the actual rotation. This task can be solved within the framework of visual servoing, using any of the solution approaches proposed in the past, as in [11] [12] [10].

The tracking loop halts when either the detection module fails to recognize the target in the image over a certain number of successive frames, or the filter has reached a consistent estimate of the target's motion parameters (i.e. the error covariance of the state estimate has converged stochastically to a constant value), which will usually occur after a finite amount of time, provided that the filter does not diverge, due to large modeling errors and/or roundoff errors or lack of observability [2] [1]. Bar-Shalom, for example, uses two hypothesis tests on the innovations to determine when the filter has reached a consistent steady state to within some confidence level [1].

4. Planning Module

This module implements the high-level multiplexing of the camera among multiple targets moving anywhere within the camera's field of regard. Specifically, it shall maintain a set of candidates, the *tracking set*, and select one target at a time from this set to pass on to the control/tracking module.

A first-order dynamic model is used to represent target motion. The parameters of this model are estimated by the control/tracking module while the target is being tracked, as was explained in the previous section. The planning module subsequently uses this model to predict forward the target's motion trajectory, until the target is scheduled again for tracking.

The main objectives in the design of this module are to maximize the *degree of multiplexing*, defined as the size of the tracking set, and minimize the target loss rate, i.e. maintain any one target in the tracking set for as long as possible (without losing it).

The next section describes our implementation of the planning module algorithm, which is also illustrated via a flowchart in Figure 1.

4.1. The Algorithm

The following points highlight the main aspects of how the algorithm works.

- The tracking set is initially empty. Tracking candidates, consisting of newly detected targets, are inserted into this set during the course of the algorithm. The initial position, time of detection and an appearance description are assumed to be specified (by the external detection mechanism) for each candidate.

- Each target in the tracking set is characterized by two parameters: the maximum tracking time, denoted by τ, and the maximum inter-tracking time, denoted by δ. The former specifies the maximum time that the control/tracking module is allowed to spend on the target, while the latter specifies the maximum time between consecutive trackings of the target.

- Targets in the tracking set are stored in a priority queue in order of *expiration time*, defined as δ time units from when the target was last tracked. A target that

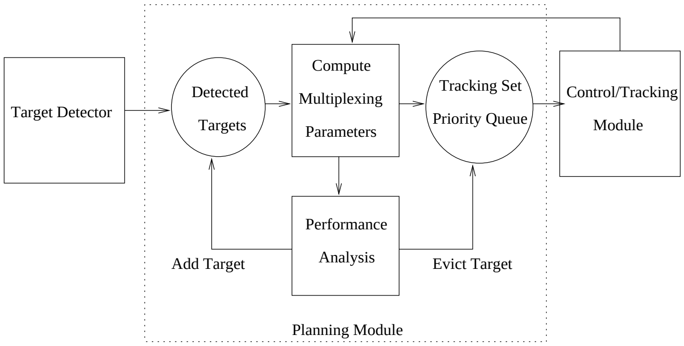

Figure 1. Flow chart of planning module operation.

is not scheduled on or before its pre-computed expiration time is presumed lost, and is immediately removed from the tracking set.

- The main loop of the algorithm operates as follows. The first non-expired target in the priority queue is passed on to the control/tracking module to be tracked for a maximum amount of time defined by the value of the target's τ. If successful in acquiring and tracking the target, the control/tracking module returns an updated estimate of its 3D motion parameters, and an associated covariance matrix. These results are used to update the target's parameters τ and δ. Finally, the tracking set is augmented or reduced by one target based on a performance analysis of the system.

4.2. Queuing-based Performance Evaluation

This is the adaptive component of the planning module algorithm that monitors system performance and determines how to adjust the workload, i.e. the tracking set, to improve it. To this end, we model the system with the single server queue shown in Figure 2, and use queuing theory framework to determine its performance [8] [9].

In this model, customers correspond to targets, the service center represents the control/tracking module, and the delay center the planning module. Each customer is delayed for a certain period of time, called the *think time* in queuing theory terminology, then proceeds to the service center where it waits in the server queue for its turn. Each customer is serviced for no longer than a pre-specified time, called its *service requirement.* Obviously, these two parameters correspond to the target parameters δ and τ. The think time and service requirement of all the customers in the queuing system collectively define its workload, and its performance is computed as a function of this workload. As such, useful performance indicators of our system can hence be computed based on the targets' δ and τ parameters, namely:

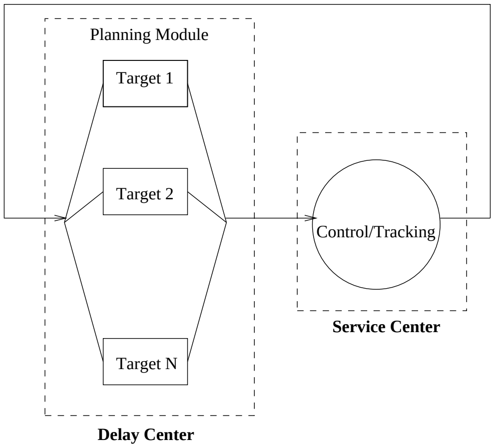

Figure 2. System Queueing Model.

- *Utilization:* $U = \sum_{c=1}^{c=N} \frac{\tau_c}{\tau_c + \delta_c}$
- *Throughput:* $X = \sum_{c=1}^{c=N} \frac{1}{\tau_c + \delta_c}$

where τ_c and δ_c are the think time and service requirement of the c-th customer, respectively.

4.3. Computation of Target Parameters

The planning algorithm above is essentially a scheduling strategy that allocates targets in time slots, one target per slot. Two parameters are instrumental to this allocation scheme; the length of a target's time slot, and the time between any two consecutive slots allocated to a target, which respectively correspond to the two target parameters τ and δ.

As explained section 3, tracking a target requires that the Kalman filter at least reach a consistent state within a given level of confidence. The length of time this might take is a random variable, and is not easily determined a priori, without actually running the Kalman filter. Therefore we simply estimate τ empirically based on observed values of the actual tracking time (i.e. the actual time that the Kalman filter took to reach a consistent state), denoted τ_a.

The inter-tracking time δ is determined heuristically as a function of the error covariance of the predicted target position as follows. During the time that a target is not being tracked, its motion is predicted forward based on its motion model. However, because the state prediction error accumulates over time, we need to determine the maximum inter-tracking time δ such that this error is not 'too large' as to impede target re-acquisition. For this, we developed a heuristic method that computes for any δ the probability that the target will lie inside the field of view centered at the predicted 3D position of the target. Then we used a binary search with objective function being this probability. The assumption here of course is that this function is strictly monotonous.

5. Experiments

To test the design and performance of our system, we designed simulation experiments based on synthetic targets moving on a plane. The various computational modules proposed in this paper were implemented in C++, except for the vision sub-module, which we simulated as a 'black box' with a certain failure rate (i.e. failure to detect the target when it is indeed in the field of view). This experimental framework can serve as a design tool of real implementations of the system. By setting the proper values of the various control parameters to match the surveillance situation of interest, the framework can be used to predict the performance of the system.

To illustrate, we tested the system within a typical urban surveillance setting, as a function of camera turn delay and video frame rate. Figure 3 below shows the results for one performance measure, the ratio of the total tracking of a target and the total time the target has remained in the camera field of regard.

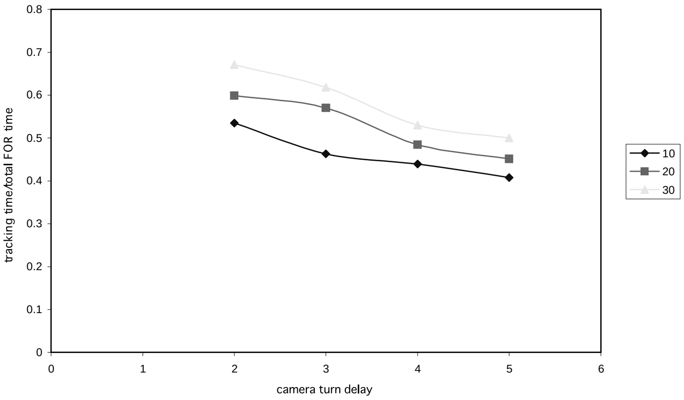

Figure 3. Ratio of total tracking time as a function of frame rate and camera turn delay.

6. Conclusions

This article presented the design of a novel surveillance system for tracking multiple targets in a wide area using a single narrow-field-of-view camera. The system so far has only been tested off-line on synthetic data. In the future, we aim to extend the current implementation of the system to handle real-time operation. Alternative, more robust heuristics for computing the target multiplexing parameters will also be investigated.

References

[1] Y. Bar-Shalom and T. E. Fortmann, *Tracking and Data Association*, Academic Press, New York, NY, 1988.

[2] R. G. Brown and P. Y. C. Hwang, *Introduction to Random Signals and Applied Kalman Filtering*, John Wiley & Sons, New York, 1983.

[3] I. Cohen and G. Medioni, “Detecting and Tracking Moving Objects for Video Surveillance,” in *IEEE Proc. Computer Vision and Pattern Recognition*, June 1999.

[4] I. J. Cox, “A Review of Statistical Data Association Techniques for Motion Correspondence,” *International Journal of Computer Vision*, Vol. 1, No. 10, pp. 53–66, 1993.

[5] A. Gelb, *Applied Optimal Estimation*, MIT Press, Cambridge MA, 1974.

[6] I. Haritaoglu, D. Harwood, and L. S. Davis, “W4S: A Real-Time System for Detecting and Tracking People in 21/2 D,” in *Proc. Europeen Conf. Computer Vision*, June 1998.

[7] S. Haykin, *Adaptive Filter Theory*, Prentice Hall, Englewood Cliffs, New Jersey, 1986.

[8] R. Jain, *The Art of Computer Systems Performance Analysis Techniques for Experimental Design, Measurement, Simulation, and Modeling*, John Wiley & Sons, 1991.

[9] E. Lazowska, J. Zahorjan, G. S. Graham, and K. Sevcik, *Quantitative System Performance*, Prentice-Hall, Inc, Englewood Cliffs, New Jersey, 1984.

[10] T. Matsuyama, “Cooperative Distributed Vision - Dynamic Integration of Visual Perception, Action, and Communication -,” in *Proc. Int. Conf. Computer Vision*, 1998.

[11] N. Papanikolopoulos, P. K. Koshla, and T. Kanade, “Vision and Control Techniques for Robotic Visual Tracking,”

[12] e. a. Wixson, L., “Image Alignment for Precise Camera Fixation and Aim,” in *Proc. Conf. Computer Vision and Pattern Recognition*.