

On-Line Control of Active Camera Networks for Computer Vision Tasks

Adrian Ilie and Greg Welch
Department of Computer Science
University of North Carolina at Chapel Hill
Chapel Hill, NC 27599-3175, USA
Email: {adyilie,welch}@cs.unc.edu

Abstract—Large networks of cameras have been increasingly employed to capture dynamic events for tasks such as surveillance and training. When using active cameras to capture events distributed throughout a large area, human control becomes impractical and unreliable. This has led to the development of automated approaches for on-line camera control. We introduce a new automated camera control approach that consists of a *stochastic performance metric* and a *constrained optimization method*. The metric quantifies the uncertainty in the state of multiple points on each target. It uses state-space methods with stochastic models of the target dynamics and camera measurements. It can account for static and dynamic occlusions, accommodate requirements specific to the algorithm used to process the images, and incorporate other factors that can affect its results. The optimization explores the space of camera configurations over time under constraints associated with the cameras, the predicted target trajectories, and the image processing algorithm. The approach can be applied to conventional surveillance tasks (e.g., tracking or face recognition), as well as tasks employing more complex computer vision methods (e.g., markerless motion capture or 3D reconstruction).

Index Terms—camera control; active camera networks; computer vision; surveillance; motion capture; 3D reconstruction

I. INTRODUCTION

Many computer vision applications, such as motion capture and 3D reconstruction of shape and appearance, are currently limited to relatively small environments that can be covered using fixed cameras with overlapping fields of view. There is demand to extend these and other approaches to *large environments*, where events can happen in *multiple dynamic locations*, simultaneously. In practice, many such large environments are *sporadic*: events only take place in a few *regions of interest* (ROIs), separated by regions of space where nothing of interest happens. If the locations of the ROIs are static, acceptable results can be obtained by straightforward replication of static camera setups used for small environments. However, if the locations of the ROIs are dynamic, coverage needs to be ensured throughout the entire volume. Using an increasing number of fixed cameras is impractical due to concerns over increased requirements in terms of computation and monetary cost, bandwidth and storage.

One practical solution to this problem is using active cameras to cover sporadic environments. Active cameras have been used in surveillance [1] and computer vision fields such as motion capture [2] and robotics [3]. What makes them

so versatile is their capability to change their pan and tilt settings to aim in the direction of dynamic ROIs, and zoom in or out to best enclose the ROIs in their field of view. However, this versatility comes at a cost: in order to capture dynamic events, active cameras need to be controlled on-line, in real-time. Control decisions need to be made as events are happening, and to take into account *factors* such as target dynamics and camera capabilities, as well as *requirements* from the computer vision algorithms the images are captured for, such as preferred camera configurations, capture durations and image resolutions.

We present an approach that controls a network of active cameras on-line, in real-time, such that they capture multiple events taking place simultaneously in a sporadic environment and produce the best possible images for processing using computer vision algorithms. We approach camera control as an *optimization problem* over the space of possible *camera configurations* (combinations of camera settings) and over time, under *constraints* derived from knowledge about the cameras, the predicted ROI trajectories and the computer vision algorithm the captured images are intended for. Optimization methods rely on objective functions that quantify the “goodness” of a candidate solution. For camera control, this objective function is a *performance metric* that evaluates dynamic, evolving camera configurations over time.

The rest of the paper is organized as follows. In Section II we present a few performance metrics and camera control methods. Section III details our performance metric, and Section IV describes our control method. In Section V we present simulated experimental results. We discuss some future work and conclude the paper in Section VI.

II. PREVIOUS WORK

A. Performance Metrics

Several researchers have attempted to express the intricacies of factors such as placement, resolution, field of view, focus, etc. into metrics that could measure and predict camera performance in computer vision algorithms.

Wu et al. [4] estimate the uncertainty in multi-camera stereo using the 2D quantization error. Chen [5] proposes a metric that takes into account resolution and occlusion for motion capture. Olague and Mohr [6] analyze the propagation of uncertainty in 3D reconstruction, and use the maximum value

of the covariance matrix as an optimization criterion. Denzler et al. [7] derive a performance metric based on conditional entropy to select the camera parameters that result in sensor data containing the most information for the next state estimation. Allen [8] introduces steady-state uncertainty as a performance metric for optimizing the design of multi-sensor systems. In previous work [9] we illustrate the integration of several performance factors into this metric and envision applying it to 3D reconstruction using PTZ cameras.

B. Camera Control Methods

Camera control methods fall mainly into two broad categories: adaptation of scheduling policies, algorithms and heuristics from other domains, and optimization of a quality metric. We lists a few examples from each category below.

Qureshi and Terzopoulos [10] propose a virtual testbed for surveillance algorithms and use it to demonstrate two adapted scheduling policies: first come, first serve (FCFS) and earliest deadline first (EDF). Costello et al. [11] present and evaluate the performance of several scheduling policies in a master-slave camera configuration. Camera scheduling is cast as a multi-class scheduling problem, with class assignments done based mainly on the number of times a target has been observed. The authors of [12], [13] cast on-line camera saccade planning for acquisition of high resolution images into a kinetic traveling salesperson problem. Bakhtari et al. [14] use principles from vehicle dispatching for tracking and state estimation of a single target with 4 PTZ cameras and a static overview camera. The goal is to select the optimal camera subset for data fusion and maneuver them in response to target motion while keeping other cameras available for future demands. Lim et al. [15] introduce the concept of task visibility intervals (TVIs) and propose solving the camera scheduling problem using dynamic programming and greedy heuristics.

Yous et al. [16] propose a camera assignment scheme based on the visibility analysis of a coarse 3D shape produced in a preprocessing step to control multiple Pan/Tilt cameras for 3D video of a moving object. The optimization is then extended into the temporal domain to ensure smooth camera movements. Krahnstoeber et al. [17] present an approach to schedule 4 PTZ cameras for a biometric task. Multiple candidate schedules are constructed as sequences of targets to be covered by each camera, then evaluated using a probabilistic performance objective function to optimize the success probability of the biometric task. Broaddus et al. [18] present *ACTvision*, a system consisting of a network of PTZ cameras and GPS sensors covering a single connected area that aims to maintain visibility of designated targets. They use a joint probabilistic data association algorithm to track the targets. Cameras are tasked to follow specific targets based on a cost calculation that optimizes the task-camera assignment and performs hand-offs from camera to camera.

In [19], Denzler et al. present a performance metric for selecting the optimal focal length in 3D object tracking. The determinant of the a posteriori state covariance matrix is

used to measure the uncertainty derived from the expected conditional entropy given a particular action. Visibility is taken into account by considering whether observations can be made and using the resulting probabilities as weights. Optimizing this metric over the space of possible camera actions yields the best actions to be taken by each camera. Deutsch et al. [20], [21] improve the process by using sequential Kalman filters to deal with a variable number of cameras and occlusions, predicting several steps into the future and speeding up the computation. Sommerlande and Reid add a Poisson process to model the potential of acquiring new targets by exploring the scene [22], examine the resulting camera behaviors when zooming [23], and evaluate the effect on the performance of random and first-come, first-serve (FCFS) scheduling policies [24].

III. PERFORMANCE METRIC

We define the performance of a camera configuration as its ability to resolve 3D features in the working volume, and measure it using the uncertainty in the state estimation process. We use state-space models [25] to describe target dynamics and measurement systems. Formally, at time step t , the system state is described by a *state vector* $\bar{x}_t \in \mathbb{R}^n$ which may include elements for position, orientation, velocity, etc. Given a point in the state space, a mathematical *motion model* can be used to predict how the target will move over a given time interval. Similarly, a *measurement model* can be used to predict what will be measured by each sensor. We measure the uncertainty in the state \bar{x}_t using the a posteriori error covariance P_t^+ , which we compute by applying the Kalman Filter [26] equations to elements of the state-space models. The Kalman filter consists of a *predict step* that projects the a priori state \bar{x}_t^- and error covariance P_t^- ahead in time, and a *correct step* that computes the Kalman gain and updates the state and error covariance estimates to \bar{x}_t^+ and P_t^+ using the available measurements \bar{z}_t .

In [9], we introduced the concept of *surrogate models* to allow evaluation of the metric in state-space only where needed: at a set of 3D points associated with each ROI. The metric values can be aggregated over the points in the surrogate model of each ROI and over the entire environment. Due to the dynamic nature of the events being captured and the characteristics of the active cameras used to capture images, spatial aggregation of metric values for the current camera configuration is not sufficient. Future camera configurations need to be evaluated as well, which requires temporal aggregation. Our performance metric evaluates a *plan*: a temporal sequence of camera configurations up to a *planning horizon*. The difficulty is that at each time instant a camera's measurement can be successful or unsuccessful. Denzler et al. [7] introduced a way to deal with visibility at each step. Deutsch et al. [20] extended this approach to multiple steps into the future using a visibility tree, and then sped up the evaluation by linearizing the tree and extended the approach to multiple cameras using a sequential Kalman filter in [27]. We employ a similar approach, but use a norm of the error covariance P_t^+

instead of entropy as the performance metric, and aggregate uncertainties over the entire horizon instead of simply taking the last uncertainty value. This avoids penalizing plans with no captures at the end of the planning horizon, where the absence of measurements leads to high uncertainties.

The metric computation works in tandem with a separate Kalman filter that is used to estimate the ROI trajectories. At each time instant, this filter incorporates the latest measurements from cameras and other sensors, and saves its *current estimate* (x_0^-, P_0^-). This estimate is the starting point for all metric evaluations. We compute the performance metric for each candidate plan by repeatedly stepping forward in time until the planning horizon, while applying the Kalman filter equations and changing relevant state-space model parameters at each time step. We use the motion models to predict ROI trajectories, and update the measurement models with the camera parameters corresponding to the configurations planned for each time step. Since future measurements are not yet available, we use the predicted ROI trajectories to generate predicted measurements instead. The Kalman correct step is evaluated sequentially [21] for all sensors involved. We perform aggregation over space and time using weighted sums, with weights quantifying the relative importance of elements at various levels, such as points in a ROI’s surrogate model, ROIs, or time instants. Equation 1 illustrates the general formula for the metric computation using weighted sums.

$$\mathcal{M} = \sum_{r=1}^{nROIs} u_r \sum_{p=1}^{N_r} v_p \sum_{t=1}^H w_t \sqrt{\max(\text{diag}_{pos}(P_t^+))} \quad (1)$$

$nROIs$ is the number of ROIs, N_r is the number of points in the surrogate model of ROI r , H is the planning horizon. u_r , v_p and w_t are relative weights for each ROI r , model point p and time step t , respectively. We use the square root of the maximum value on the diagonal of the portion of the error covariance matrix P_t^+ corresponding to the position part of the state to obtain a single number. Other functions such as the determinant or the trace can be used instead of the diagonal maximum. One benefit of using the square root of the covariance is that the measurement unit for the metric is the same as the measurement unit of the state space. For example, if the state consists of 3D point positions measured in meters, the metric value will also be in meters. This makes it more intuitive for a system user to specify application requirements such as the desired maximum error in a particular area where important events take place.

Using state-space models as our framework for computing the performance metric has several advantages. One is the possibility to seamlessly integrate measurements from heterogeneous sensors such as cameras and GPS to estimate the state. Another is that it allows easy incorporation of performance factors and algorithmic requirements. In [9], we have detailed the incorporation of several factors into a stochastic performance metric: object distance, incidence angle, image resolution, field of view, occlusion, focus distance, depth

of field and camera selection. Small changes to the metric allow it to adapt to various computer vision algorithms. For example, for target tracking and motion capture, a system designer only needs to appropriately choose the components of the state vector. Surveillance tasks such as following a target or capturing face images for biometric purposes can be implemented by tuning the relative importance of the target’s surrogate model, and of the points corresponding to the face, respectively. Surrogate models can be adapted for the requirements of various 3D reconstruction algorithms: models used in stereo reconstruction could include local surface orientation, and models used in volumetric reconstruction could consist of medial axis-like representations, instead of 3D points. Finally, our metric naturally incorporates trade-offs such as the one between uninterrupted, long captures at larger fields of view and lower resolution vs. short captures at narrow fields of view and higher resolution, punctuated by transitions.

IV. CONTROL METHOD

We define optimization in active camera control as the exploration of the space of possible solutions in search for the best solution as evaluated by the performance metric. We found that exploring all possible camera settings combinations is intractable. Instead, we explore the space of camera-ROI assignments, and compute the best settings corresponding to each assignment using geometric reasoning: the best results are usually obtained when the ROI trajectories are enclosed in the camera fields of view as tightly as possible.

The optimization process first predicts the ROI trajectories, then uses them to construct and evaluate a number of candidate plans for each camera. A plan consists of a number of *planning steps*, which in turn consist of a *transition* (during which the camera changes its settings) and a *dwelling* (during which the camera captures, with constant settings). Candidate plans differ in the number and duration of planning steps up to the planning horizon.

Unfortunately, evaluating all possible combinations of candidate plans for all cameras is still intractable. We investigated multiple ways of reducing the search space size, of which we found the most effective to be decomposing the optimization problem into subproblems and solving each subproblem independently. Our implementation runs on-line, in real-time, and consists of two components: centralized *global assignment* and distributed *local planning*.

A. Global Assignment

The global assignment component accomplishes two tasks: grouping ROIs into *agencies* and assigning cameras to each agency. We create agencies by clustering together ROIs that are close to each other and predicted to be heading in the same direction. We use predicted ROI trajectories to cluster the ROIs into a minimum number of non-overlapping clusters of a given maximum diameter. ROI membership needs to exhibit hysteresis to help keep the assignments stable: when an agency’s membership changes, all cameras assigned to it and other cameras nearby need to have their current plans

reevaluated. If these evaluations result in changes in plans or assignments, cameras may need to transition, and end up spending less time capturing, which usually results in worse performance. To avoid this problem, we use a *minimal change* heuristic, consisting of three steps:

- 1) if any agency has become too large, iteratively remove ROIs from it until it becomes small enough;
- 2) assign all unassigned ROIs to the agencies closest to them; form new agencies for isolated ROIs;
- 3) if any two agencies are close enough to each other, merge them into a single agency.

We use a greedy heuristic to assign cameras to each agency, based on their potential contribution to it. Only *available* cameras are taken into account. A camera is considered available if its state for the next planning cycle is the start of a transition or an occlusion. Additionally, a camera's dwell can be interrupted if it has lasted longer than a specified minimum duration. The heuristic iteratively tries assigning all available cameras to all agencies, searching for the camera-agency assignment that best improves the metric value for the agency. Improvement is measured using the ratio between the metric value before and after making the assignment. The resulting plans are compared with plans obtained by prolonging the current plans up to the planning horizon whenever possible, and the greedy assignments are only applied if they perform better.

The plans corresponding to each camera-agency assignment are generated heuristically by assuming the worst-case scenario: the camera is repeatedly set to transition, then capture for as long as possible, with a field of view as wide as possible. While other scenarios, in which the camera captures for shorter intervals, may result in better performance by using tighter fields of view, the goal of this heuristic is only to help assess the potential contribution a camera can have to the capture of the ROIs in the agency it is being assigned to. The heuristic also takes into account predicted static and dynamic occlusions, and plans transitions during occlusions whenever possible, in order to minimize the time when the camera is not capturing. Static occlusions are precomputed off-line for each camera. Dynamic occlusions are computed on-line, using the predicted ROI trajectories.

B. Local Planning

Local planning at the level of each agency is concerned with the locally-optimal capture of the ROIs in the agency. All cameras assigned to each agency capture all member ROIs, and no further camera-ROI assignment decisions are made at this level. The planning decisions made at this level are on when and for how long each camera should dwell (capture), and when it should transition to a new configuration.

The current configuration of each camera, corresponding to the first step in a plan, is arguably the most important: configurations corresponding to subsequent steps are revised during subsequent optimization cycles, when the planning method can take advantage of the most recent predictions of ROI trajectories. We call the resulting planning process *myopic*: exhaustive exploration of all possible dwell durations

is only done during the first planning step, and heuristic computations are used for subsequent steps up to the time horizon. The heuristic is the same one employed for generating plans during global assignment. Myopic planning can still result in too many plans to evaluate for running in real-time. We reduce the number of candidate plans by planning transitions during occlusions and limiting how many planning steps can fit before the planning horizon.

All possible combinations of candidate plans for all cameras are explored exhaustively using backtracking. To achieve on-line, real-time control, the set of candidate plans is sorted so that the most promising plans are evaluated first. We use prior experimental observations to derive criteria for judging a plan's potential. While not a guarantee that the best plan would be chosen on time, we have found this heuristic to closely approximate an exhaustive search.

V. EXPERIMENTAL RESULTS

We used data from a training exercise to test how a prototype implementation of our method can control 6 PTZ cameras to observe 6 ROIs, each ROI enclosing a single exercise participant. The exercise was performed by members of the United States Marines Corp (USMC), and captured on-site at the Sarnoff Corporation in Princeton, NJ. We used the game engine-based simulator from [28] to run multiple simulations using the same input data. The simulator provides means of controlling a number of virtual cameras in a simulated environment and retrieving images from them. We extended the simulator's functionality by providing means for an outside application to control the virtual characters in the scene.

The exercise lasted 170 seconds, and our method performed a complete optimization every second. The exercise scenario was for a squad of 4 Marines to patrol, cross a danger zone between 2 buildings, perform cordon searches on a civilian, neutralize a sniper threat, and move out to secure an area. Figure 1 shows a top-down view of the exercise site (rotated 90° counterclockwise).

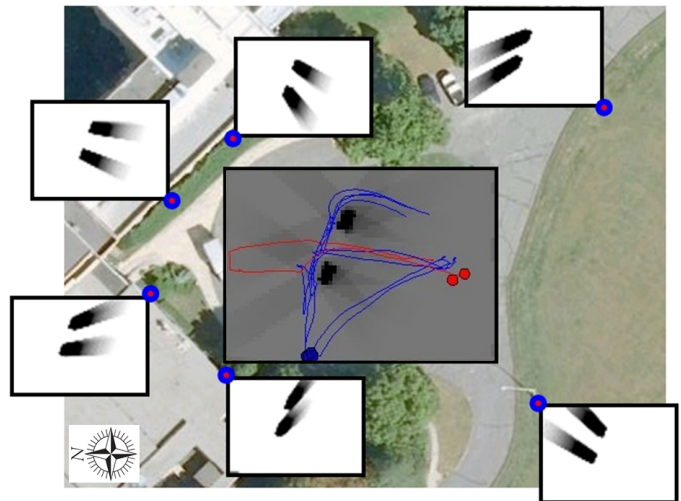


Figure 1. Top-down view of the exercise site.

Marine tracks are shown in blue, civilian and sniper tracks are shown in red. We placed the 6 virtual cameras as follows: 4 cameras along the building walls and 2 cameras on light poles in the parking lot. We precomputed static occlusions for each camera using the simulator’s ability to provide ground truth for the virtual characters in the camera images (silhouettes, bounding boxes, total and visible pixel counts) and stored the results in visibility maps. Camera locations are shown as blue circles, each accompanied by its visibility map. Brighter values denote higher visibility. The aggregated visibility is also shown as a grayscale image overlaid on a top-down satellite view of the site and aligned to match the area where the exercise took place. There are also 2 shipping containers which served as props during the exercise and helped us test the impact of static occlusions. They appear as dark spots of zero visibility in the aggregated visibility image, and cast “shadows” in the visibility maps.

During the exercise, target trajectories were captured as GPS measurements over time. The approach from [18] was used to capture images from 2 PTZ cameras and refine the trajectory estimates. We filtered the trajectory data to reduce noise, sampled it at every second and used the samples as input for our prototype implementation. During experiment runs, we used the 3D points on these input trajectories to generate simulated 2D measurements in each virtual camera and 3D (long., lat., alt.) simulated GPS measurements. We added noise (precomputed for repeatability) to each measurement before incorporating it into the performance metric computation. The surrogate model for each exercise participant consisted of 2 cubic regions, 1 meter on the side, stacked on the vertical axis. The coordinates of the center of each region were included in the state, and a PV state model [25] was used in the Kalman filter-based performance metric evaluation process.

We ran multiple simulations using the same input data and tuning the parameters of our control method. The results we obtained led to the observations summarized below:

- 1) Using active cameras always performed better than using fixed cameras (simulated by zooming out the active cameras and aiming them at the center of the scene). The positive effect of higher resolution when controlling the cameras was more than enough to compensate for the interruptions in capture due to transitions.
- 2) Decreasing the frequency of GPS measurements made the control method zoom out the cameras to compensate, which resulted in worse performance.
- 3) Increasing the visibility threshold led to more occlusions, resulting in plans with captures that were more fragmented by transitions, but the increased image resolution compensated for some of the performance loss.
- 4) Increasing the length of the planning cycle to allow for more comprehensive searches resulted in fewer disruptions in camera membership for each agency, but also in an increased number of times when wrong trajectory predictions led to the ROIs not being captured.
- 5) Increasing the planning horizon length and the clustering diameter led to the heuristic generating longer captures

at larger fields of view and lower resolution, which resulted in worse performance.

One important result was that our planning heuristics can produce satisfactory results and serve as a fall-back when the search space is too large to be explored in real-time. We varied the number of candidate plans explored by the local planning component of our method between *heuristic* (using the plan generated heuristically by the global assignment), *selective* (only explore a small number of plans) and *exhaustive* (explore all candidate plans when possible — 151 out of 170 planning cycles). Figure 2 shows the results, including the metric values when using static cameras, shown for comparison.

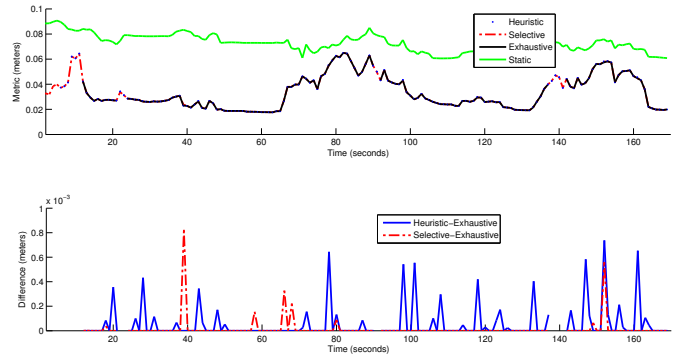


Figure 2. Metric over time when varying the number of plans explored. *Top:* Metric values over time for various local exploration methods.

Bottom: Differences in metric values over time.

As expected, the metric values in the case of static cameras are significantly higher, but part of the price paid for the lower resolution is offset thanks to the fact that there are no transitions. The 3 curves corresponding to varying the number of candidate plans explored are very close to each other, to the point of being indistinguishable at the scale in Figure 2 (top). Figure 2 (bottom) shows the differences between them at a larger scale. The heuristic arrived at the same result as the exhaustive search for 109 out of 151 cycles, or 72% of the time, and the selective search did so for 143 cycles, or 94% of the time. For the remaining times when the results were different, the average difference in the metric values was 0.01m for the heuristic and 0.0087m for the selective search.

In our experiments, we have also observed desirable complex camera behaviors emerging automatically, without user intervention, as a direct result of candidate plans being generated and evaluated using the performance metric. For example, when the ROIs in a particular agency were predicted to be occluded from a camera’s view by one of the shipping containers, the camera was automatically reassigned to a different agency that contained ROIs predicted to be visible. Cameras were zoomed out to cover a larger area when participants moved faster, and zoomed back in when the ROIs slowed down. Multiple cameras also automatically had their transitions coordinated: inside the same agency, camera transitions were planned such that at any given time coverage was ensured

by at least 2 cameras simultaneously. When an agency only had 2 cameras assigned to it, the global assignment tried to assign a 3rd camera to the agency whenever transitions were needed to ensure continuing coverage.

VI. CONCLUSIONS AND FUTURE WORK

We presented a new on-line camera control approach that treats camera control as an optimization problem. For the objective function, we developed a versatile performance metric that can incorporate both performance factors and application requirements. To reduce the size of the search space and arrive at an implementation that runs in real-time, our control method breaks down the problem into subproblems. We first employ proximity and minimal change heuristics to decompose the problem into subproblems and a greedy heuristic to assign cameras to subproblems based on evaluating candidate plans. We then solve each subproblem independently, generating and evaluating candidate plans as time allows.

The simulator we used [28] closely approximates a real camera's control interface and has allowed us to repeatedly test our method with the same input data and tune its parameters. We are currently working on applying our control method to real cameras and investigating the possible use of smart PTZ cameras for distributed processing. We plan to replace the simulated camera measurements obtained from the external tracker with actual measurements from image tracking. We are also studying the impact of imprecise geometric calibration and camera settings accuracy on our control method.

ACKNOWLEDGMENTS

We acknowledge our sponsors and collaborators in the "Behavior Analysis and Synthesis for Intelligent Training (BASE-IT)" project: ONR grant N00014-08-C-0349, Roy Stripling, Ph.D., Program Manager, led by Greg Welch (PI) at UNC, Amela Sadagic (PI) at the Naval Post-graduate School, and Rakesh Kumar (PI) and Hui Cheng (Co-PI) at Sarnoff.

REFERENCES

- [1] R. T. Collins, A. J. Lipton, T. Kanade, H. Fujiyoshi, D. Duggins, Y. Tsin, D. Tolliver, N. Enomoto, O. Hasegawa, P. Burtl, and L. Wixson, "A system for video surveillance and monitoring," The Robotics Institute, Carnegie Mellon University, Pittsburgh PA, Tech. Rep. CMU-RI-TR-00-12, May 2000.
- [2] J. Davis, "Mixed scale motion recovery," Ph.D. dissertation, Stanford University, 2002.
- [3] A. J. Davison and D. W. Murray, "Simultaneous localisation and map-building using active vision," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2002.
- [4] J. J. Wu, R. Sharma, and T. S. Huang, "Analysis of uncertainty bounds due to quantization for three-dimensional position estimation using multiple cameras," *Optical Engineering*, vol. 37, pp. 280–292, January 1998.
- [5] X. Chen, "Designing multi-camera tracking systems for scalability and efficient resource allocation," Ph.D. dissertation, Stanford University, 2002.
- [6] G. Olague and R. Mohr, "Optimal camera placement for accurate reconstruction," *Pattern Recognition*, vol. 35, no. 4, pp. 927–944, April 2002.
- [7] J. Denzler, M. Zobel, and H. Niemann, "On optimal camera parameter selection in kalman filter based object tracking," in *24th DAGM Symposium on Pattern Recognition*, 2002, pp. 17–25.
- [8] B. D. Allen, "Hardware design optimization for human motion tracking systems," Ph.D. dissertation, University of North Carolina at Chapel Hill, December 2007.
- [9] A. Ilie, G. Welch, and M. Macenko, "A stochastic quality metric for optimal control of active camera network configurations for 3D computer vision tasks," in *Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications*, Marseille, France, October 2008.
- [10] F. Z. Qureshi and D. Terzopoulos, "Towards intelligent camera networks: a virtual vision approach," in *2nd Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, Beijing, China, October 2005, pp. 177–184.
- [11] C. J. Costello, C. P. Diehl, A. Banerjee, and H. Fisher, "Scheduling an active camera to observe people," in *ACM 2nd international workshop on Video surveillance & sensor networks*, 2004, pp. 39–45.
- [12] A. del Bimbo and F. Pernici, "Towards on-line saccade planning for high-resolution image sensing," *Pattern Recognition Letters*, vol. 27, no. 15, pp. 1826–1834, November 2005.
- [13] A. D. Bagdanov, A. del Bimbo, and F. Pernici, "Acquisition of high-resolution images through online saccade sequence planning," in *3rd ACM international workshop on Video surveillance & sensor networks*, 2005, pp. 121–130.
- [14] A. Bakhtari, M. D. Naish, M. Eskandari, E. A. Croft, and B. Benhabib, "Active-vision-based multisensor surveillance—an implementation," *IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews*, vol. 36, no. 5, pp. 668–680, September 2006.
- [15] S.-N. Lim, A. Mittal, and L. S. Davis, "Task scheduling in large camera networks," in *Asian Conference on Computer Vision*, 2007.
- [16] S. Yous, N. Ukita, and M. Kidode, "An assignment scheme to control multiple pan/tilt cameras for 3d video," *Journal of Multimedia*, vol. 2, no. 1, pp. 10–19, February 2007.
- [17] N. Krahnstoeber, T. Yu, S.-N. Lim, K. Patwardhan, and P. Tu, "Collaborative real-time control of active cameras in large scale surveillance systems," in *Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications*, Marseille, France, October 2008.
- [18] C. Broaddus, T. Germano, N. Vandervalk, A. Divakaran, S. Wu, and H. Sawhney, "Act-vision: active collaborative tracking for multiple ptz cameras," in *Proceedings of SPIE: Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications*, vol. 7345, Orlando, FL, USA, April 2009.
- [19] J. Denzler, M. Zobel, and H. Niemann, "Information theoretic focal length selection for real-time active 3-d object tracking," in *International Conference on Computer Vision*, vol. 1, October 2003, pp. 400–407.
- [20] B. Deutsch, M. Zobel, J. Denzler, and H. Niemann, "Multi-step entropy based sensor control for visual object tracking," *Lecture Notes in Computer Science*, vol. 3175, pp. 359–366, 2004.
- [21] B. Deutsch, H. Niemann, and J. Denzler, "Multi-step active object tracking with entropy based optimal actions using the sequential kalman filter," in *IEEE International Conference on Image Processing*, vol. 3, 2005, pp. 105–108.
- [22] E. Sommerlade and I. Reid, "Information theoretic active scene exploration," in *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, May 2008.
- [23] —, "Cooperative surveillance of multiple targets using mutual information," in *Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications*, 2008.
- [24] —, "Information-theoretic decision making for exploration of dynamic scenes," in *Proceedings of the 5th International Workshop on Attention in Cognitive Systems (WAPCV)*, May 2008.
- [25] G. Welch, B. D. Allen, A. Ilie, and G. Bishop, "Measurement sample time optimization for human motion tracking/capture systems," in *Trends and Issues in Tracking for Virtual Environments, Workshop at the IEEE Virtual Reality 2007 Conference*, G. Zachmann, Ed., Charlotte, NC, March 2007.
- [26] G. Welch and G. Bishop, "An introduction to the Kalman filter: SIGGRAPH 2001 course 8," in *Computer Graphics, SIGGRAPH 2001 course pack ed.*, ser. Annual Conference on Computer Graphics & Interactive Techniques. Los Angeles, CA, USA: ACM Press, Addison-Wesley Publishing Company, August 12-17 2001.
- [27] B. Deutsch, S. Wenhardt, and H. Niemann, "Multi-step multi-camera view planning for real-time visual object tracking," *Lecture Notes in Computer Science*, vol. 4174, pp. 536–545, 2006.
- [28] G. R. Taylor, A. J. Chosak, and P. C. Brewer, "OVVV: Using virtual worlds to design and evaluate surveillance systems," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2007, pp. 1–8.