Title

Opportunistic Image Acquisition of Individual and Group Activities in a Distributed Camera Network

Permalink
https://escholarship.org/uc/item/0x74w4bx

Journal
IEEE Transactions on Circuits and Systems for Video Technology, 27(3)

ISSN
1051-8215 1558-2205

Authors
Ding, Chong
Bappy, Jawadul H
Farrell, Jay A
et al.

Publication Date
2017-03-01

DOI
10.1109/TCSVT.2016.2593620

Peer reviewed

Opportunistic Image Acquisition of Individual and Group Activities in a Distributed Camera Network

Chong Ding, *Student Member, IEEE*, Jawadul H. Bappy, *Student Member, IEEE*, Jay A. Farrell, and Amit K. Roy-Chowdhury, *Senior Member, IEEE*

***Abstract*— The decreasing cost and size of video sensors has led to camera networks becoming pervasive in our lives. However, the ability to analyze these images effectively is very much a function of the quality of the acquired images. In this paper, we consider the problem of automatically controlling the fields of view of individual pan–tilt–zoom (PTZ) cameras in a camera network leading to improved situation awareness (e.g., where and what are the critical targets and events) in a region of interest. The network of cameras attempts to observe the entire region of interest at some minimum resolution while opportunistically acquiring high resolution images of critical events in real time. Since many activities involve groups of people interacting, an important decision that the network needs to make is whether to focus on individuals or groups of them. This is achieved by understanding the performance of video analysis tasks and designing camera control strategies to improve a metric that quantifies the quality of the source imagery. Optimization strategies, along with a distributed implementation, are proposed, and their theoretical properties analyzed. The proposed methods bring together computer vision and network control ideas. The performance of the proposed methodologies discussed herein has been evaluated on a real-life wireless network of PTZ capable cameras.**

***Index Terms*— Camera networks, distributed optimization, opportunistic sensing and tracking.**

I. Introduction

THE focus of most work in video analysis has been on improving the performance of detection, tracking, and recognition algorithms in a variety of settings. A large-scale experimental analysis has consistently demonstrated the limitations in the improvements that can be obtained by focusing solely on the processing side. Pose and image resolution remain two of the hardest challenges to be overcome for robustness in scene understanding. However, research focused on video acquisition strategies driven by the need to maximize performance goals has been quite limited.

Our goal is to develop "optimal" sensing strategies in a network of visual sensors, with a focus on tightly integrating the sensing and processing tasks in order to image, track, and identify targets effectively. We consider wide-area scene understanding problems where the sensors are pan–tilt–zoom (PTZ) cameras. In keeping with normal behavior observed widely, we assume that the scene will be populated with people acting individually or in groups. We envision that these visual sensors will have the capability to analyze their own data and perform collaborative decision making in coordination with other sensors. This would enable the sensors to maneuver themselves optimally, i.e., change the PTZ parameters of the cameras, so as to obtain image sequences that can be analyzed with a high degree of reliability. A specific challenge that this network will have to address in the process is to decide whether to image individuals separately or in groups.

A. Related Work

Historically, camera networks [1] have consisted of mostly static cameras, where the layout is designed to satisfy some predefined objective, such as covering an area. Optimal camera placement strategies proposed in [2] and [3] were solved by using a camera placement metric that captures occlusion in 3D environments. The solution to the problem of optimal camera placement given coverage constraints was presented in [4] and can also be used to come up with an initial camera configuration.

Currently, most of data collected by these networks are analyzed manually, a task that is extremely tedious and limits the potential of the installed network. This has sparked a huge interest in an automated analysis in camera networks. Tron and Vidal [5] considered distributed estimation in camera networks, while Taj and Cavallaro [6] handled distributed tracking in camera networks. A review of the automated analysis in camera networks can be found in [1]. A solution to improve performance within the realm of static cameras is to increase the total number of cameras observing a region and selectively use a subset depending on the scene [7]. Another approach is to use a system with cameras parameters that can be dynamically altered according to the scene. In [8], a master–slave configuration is used to acquire high resolution images, and scheduling is considered as a dynamic discrete optimization problem. Early work done in active vision [9] looked at the problem of moving a camera to improve imagery. Performing active vision in a distributed camera network, where the cameras coordinate among themselves, remains relatively unexplored.

Much of the research in controllable camera networks in the last decade was focused on centralized solutions to

Manuscript received December 10, 2015; revised April 2, 2016; accepted July 16, 2016. Date of publication July 20, 2016; date of current version March 3, 2017. This work was supported by NSF under Grant 1544969. This paper was recommended by Associate Editor T. Bouwmans.

C. Ding is with the Department of Computer Science and Engineering, University of California at Riverside, Riverside, CA 92521 USA (e-mail: cding@cs.ucr.edu).

J. H. Bappy, J. A. Farrell, and A. K. Roy-Chowdhury are with the Department of Electrical and Computer Engineering, University of California at Riverside, Riverside, CA 92521 USA (e-mail: mbappy@ee.ucr.edu; farrell@ee.ucr.edu; amitrc@ee.ucr.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2016.2593620

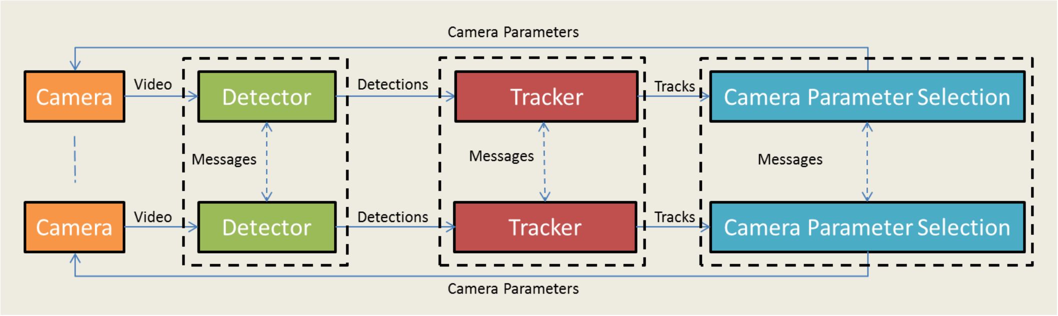


Fig. 1. Diagram depicting the framework for integrating scene analysis and PTZ control.

master–slave systems, where static cameras directed the PTZ cameras. The path planning inspired approach proposed in [10] used static cameras to track all targets in a virtual environment, while the PTZ cameras were each assigned to obtain high resolution video from a particular target. This approach showed that given the predicted tracks of all the targets, a set of one-to-one mappings between cameras and targets can be formed to acquire high resolution videos. A method for determining good sensor configurations that would maximize performance measures was introduced in [11]. The configuration framework allowed for the presence of random occluding objects. In [12], an information-theoretic approach was incorporated based on tracker uncertainty for an active scene analysis where no movement cost is considered for selecting camera parameters.

A more recent approach in [13] and [14] uses the expectation–maximization algorithm to find the optimal configuration of PTZ cameras given a map of activities, and the value of each discretized ground coordinate is determined using this map. The game theoretic approach in [15] and [16] showed how local value functions could be designed to constantly improve the tracking accuracy of all targets observed by a self-configuring camera network. Prior work in target tracking with the ability to obtain high resolution shots was shown in [17] using a fixed cost function. While the approaches in these papers were decentralized and could achieve high accuracy, they were focused largely on designing specific imaging functions on a per-target basis and not on the overall system optimization required for more complex scenes involving interactions between multiple targets. Also, the scalability of the system was limited, a fact that we address in Section III. More discussion on how to design games in general for distributed optimization can be found in [18].

In recent years, group information, such as track splitting and merging, has been studied in several tracking methods [19]–[25]. In [19], [20], and [22], group information is utilized as a constraint in order to improve the individual tracking performance. In [23], a particle filter is used to model both individual and group information jointly. Zheng *et al.* [24] and Cai *et al.* [25] addressed the group tracking problem with a descriptor of appearance features. Vezzani *et al.* [26] proposed a person reidentification method that finds the one-to-one correspondences between targets in different cameras. However, none of these works consider image acquisition with individual or group information.

1) Main Contribution: The main contribution of this paper is to present a general approach where interesting events that the targets are involved in are detected, and the system automatically decides *when, where, and how* to image these targets. An aspect that requires particular attention in this regard is that it is often difficult to maintain long tracks on objects, especially when they are interacting with each other. For example, if an individual merges with a group, we may lose track of that individual. However, if the group later splits, we may be able to get the track back. This is predicated upon the availability of high quality images at opportune moments of time so that the tracks can be reconstructed. It is the major motivation for this paper, and the experimental results presented in the following will highlight this aspect further.

B. Solution Overview

Configurations of cameras with large fields of view (FOVs) can monitor a large area of the environment but may not be able to supply images for reliable recognition tasks. Configurations where the cameras are zoomed in on specific areas of interest can gather images useful for recognition, but result in a very limited view of the scene.

In this paper, we will show how to automatically acquire high resolution images, based on the state of the scene and video surveillance system, while covering a region of interest. Ideally, the system would be able to acquire detailed images of targets before and after any events that could potentially affect the ability to create long-term tracks and recognize the targets that are being tracked.

This framework is shown in Fig. 1 and can find application in any visual surveillance system containing PTZ cameras. The raw video from each of the cameras is first processed through a detector. Any resulting detections are then associated and used to update the target tracks by the tracker. These tracks are then used to decide on the next set of PTZ settings for each camera. In a traditional completely centralized system, the raw video would be sent from each camera to a central server where the detector, tracker, and parameter selection modules would be located. In the fully distributed case, each camera would communicate only necessary information, e.g., the state estimates of the targets. The local objective function of each camera must be aligned with the global objective function, such that any local decisions, using only locally available information, improve the value of the global objective function.

TABLE I
NOTATION SUMMARY

Parameter	Variable
i-th camera	c_i
Set of all PTZ settings available to c_i	A_i
The PTZ setting for all cameras	$\mathbf{a}$
The PTZ setting for c_i, all cameras except c_i	a_i, $\mathbf{a}_{-i}$
The PTZ setting for c_i and its neighbors	$\mathbf{a}^{c_i}$
Measurement vector, measurement covariance	$\mathbf{u}$, $\mathbf{C}$
Rotation matrix from frame a to frame b	$\mathbf{R}_a^b$
State vector for track j	$\mathbf{x}_j$
Position vector for track j in world frame	$\mathbf{p}_j^w$
Area coverage utility	$U_{area}(\mathbf{a})$
Imaging utility	$U_{image}(\mathbf{a})$
Local area coverage utility	$u_{area}(\mathbf{a}^{c_i})$
Local imaging utility	$u_{image}(\mathbf{a}^{c_i})$

II. Modeling of Scene Dynamics

The purpose of this section is to define the time propagation and measurement models of our system, which will form the basis for evaluating the performance of the video analysis tasks. Additional notation is summarized in Table I. The camera network assumes that all the PTZ cameras are calibrated.

A. Time Propagation Model

With time step $T = 1$, as track j moves throughout the area, its trajectory in discrete time is modeled in state space as

$$\mathbf{x}_j(k+1) = \mathbf{\Phi}\mathbf{x}_j(k) + \boldsymbol{\omega} \tag{1}$$

where $\mathbf{x}_j = [\mathbf{p}_j^w, \mathbf{v}_j^w]^\top$ with $\mathbf{p}_j^w = [x_j^w, y_j^w, z_j^w]^\top$ is the position in the world frame, $\mathbf{v}_j^w = [v_{x,j}^w, v_{y,j}^w, v_{z,j}^w]^\top$ is the velocity in the world frame, and $j = 1, \ldots, N_t$ is the track number. Also, $\boldsymbol{\omega} \sim \mathcal{N}(\mathbf{0}_{6\times1}, \mathbf{Q})$ is the process noise and

$$\mathbf{\Phi} = \begin{bmatrix} \mathbf{I} & \mathbf{I} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \tag{2}$$

is the state transition matrix, where $\mathbf{0}$ and $\mathbf{I} \in \Re^{3\times3}$. The state estimate and its error covariance matrix are propagated between sampling instants using recursive Kalman filter equations [27].

B. Measurement Model

This section derives the measurement model for track j by camera i. The position of the jth track in the ith camera frame is $\mathbf{p}_j^{c_i} = [x_j^{c_i}, y_j^{c_i}, z_j^{c_i}]^\top$, the rotation from world to the ith camera frame is denoted by $\mathbf{R}_w^{c_i}$, and the position of the camera in the world frame is $\mathbf{p}_{c_i}^w$.

The position of track j in the ith camera frame is related to its position in the world frame by

$$\begin{bmatrix} \mathbf{p}_j^{c_i} \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_w^{c_i} & \mathbf{0} \\ \mathbf{0}^\top & 1 \end{bmatrix} \begin{bmatrix} \mathbf{I} & -\mathbf{p}_{c_i}^w \\ \mathbf{0}^\top & 1 \end{bmatrix} \begin{bmatrix} \mathbf{p}_j^w \\ 1 \end{bmatrix} \tag{3}$$

where $\mathbf{0} \in \Re^{3\times1}$ and $\mathbf{I} \in \Re^{3\times3}$ is the identity matrix.

Using the homogeneous coordinate representation, the ith camera's image plane coordinates for the jth track can be represented as $\mathbf{p}_j^i$. The result of performing feature detection on the image from Camera i is a 2D projection of the centroid position of the jth track on the image plane in pixel coordinates $\mathbf{u}_j^i$ and its associated error covariance $\mathbf{C}_j^i$. Accounting for noise, the measurement from the ith camera can be modeled as

$$\mathbf{u}_j^i = h(\mathbf{p}_j^i) + \boldsymbol{\eta}_j^i = \begin{bmatrix} \dfrac{x^i}{z_j^{c_i}} \\ \dfrac{y^i}{z_j^{c_i}} \end{bmatrix} + \boldsymbol{\eta}_j^i \tag{4}$$

where we assume that $\boldsymbol{\eta}_j^i \sim \mathcal{N}(\mathbf{0}_{2\times1}, \mathbf{C}_j^i)$. The linearized measurement model for every measurement at time step k is thus given by

$$\mathbf{u}_j^i(k) = \mathbf{H}_j^i(k)\mathbf{x}_j^i(k) + \boldsymbol{\eta}_j^i(k). \tag{5}$$

where $\mathbf{H}_j^i$ is the observation matrix [28].

C. Detection, Association, and Tracking

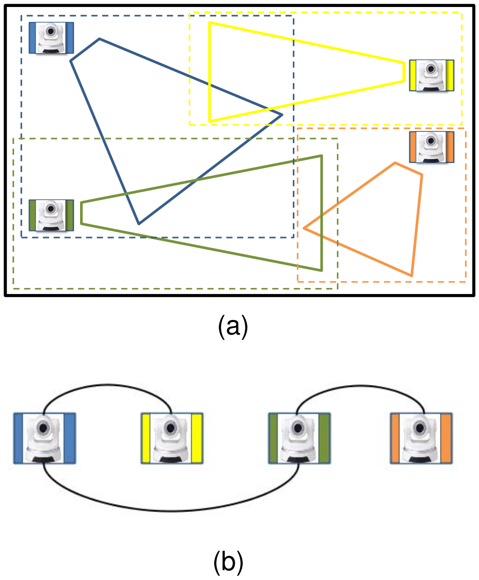
In our framework, we assume that the detector for each camera provides a centroid position $\mathbf{u}_j^i$ and error covariance $\mathbf{C}_j^i$ on the image plane for each detection. The resulting detections can then be associated with existing tracks using any existing data association technique. A review of object tracking methods can be found in [27]. It will typically be the case that each camera only detects some of the targets.

There are several different events that affect our tracker. When the probability that a detection is associated with an existing track is below a threshold, a new track is initialized by transforming the centroid position and error covariance of the detection to the world frame as the targets are on the ground plane. Since the relation between the image and ground planes is nonlinear, we use (5) to perform the measurement updates for the state estimates and error covariances for tracks using an extended Kalman filter.

III. Distributed Solution

One of the challenges in vision networks is scalability with the size of the network. This motivates us to provide a distributed solution to the problem, where each camera is able to communicate with its neighbors, as defined by a communication graph described in the following. In the proposed distributed network, the cameras are only able to communicate directly with their immediate neighbors on the communication graph and only have access to information provided by these neighbors. A major benefit of such a framework is that new cameras with varying capabilities can be easily added or removed from the network at any time. We do not address the problem of distributed tracking and data association, as these have been presented in [29]–[31], which can be used directly.

In a distributed camera network, the camera communications are modeled by a communication graph. To ease computation, we quantize the entire area B into a set of blocks, $B = \{b_1, b_2, \ldots, b_{N_b}\}$. $B^i \subseteq B$ is the set of blocks for which there exist a valid parameter setting so that the block is in the FOV of camera c_i.


(a)

(b)

Fig. 2. (a) FOV for four cameras as closed trapezoids. The dotted rectangles show the possible area that can be covered given all the settings available to each camera. (b) Corresponding vision graph, connecting cameras can have overlap in their FOVs.

Definition 1: The communication graph has cameras as nodes and there exists an edge from camera i' to camera i'' when those cameras can directly receive information from each other.

Definition 2: In vision graph, cameras are represented as nodes. The graph has an edge from camera i' to camera i'' when the set of blocks $B^{i'} \cap B^{i''} \neq \emptyset$.

These graphs are important to our problem as the communication graph determines *who knows what and when*, while the vision graphs determines which cameras can share operation regions (i.e., have overlapping FOVs when their PTZ parameters are appropriately chosen). The example in Fig. 2(a) shows four cameras each covering a portion of the area. For the regions B^i shown in Fig. 2(a), the corresponding vision graph is shown in Fig. 2(b).

A. Design of Local Utility Function

In a distributed camera network, one possible solution is to design the objective functions so that they implement a potential game [18], which guarantees that the cameras will converge to a Nash equilibrium (NE) (see Section III-D for the proof). While this is a distributed solution, there are a few undesirable characteristics such as allowing only one camera to change its proposed parameter settings at each iteration step [16]). The existing literature only accounts for the communication graph connectivity and the amount of communication increases with the number of cameras. In the following, we incorporate information from the vision graph by considering the cameras that can affect the value of the local camera's PTZ settings.

We present a design of the value functions for the different objectives of our system. In the following sections, let $C_i \subseteq C$ be the set of cameras containing c_i and its neighbors in communication graph, and $A^{c_i} = \times_{i \in C_i} A_i$ be the set of possible PTZ settings of the cameras in C_i.

1) Local Area Coverage: Instead of considering all cameras in the network, we write the local area coverage utility to only consider the PTZ settings $\mathbf{a}^{c_i} \in A^{c_i}$ as

$$u_{\text{area}}(\mathbf{a}^{c_i}) = \sum_{p \in B_j} \left(1 - \prod_{q \in C_i} (1 - \beta_p^q)\right) \tag{6}$$

where B_j is the set of blocks observed by the cameras in C_i and β_p^q is defined as

$$\beta_p^q = \begin{cases} 1 - e^{-\lambda_1 \frac{r_p^q}{r_p^{\max}}}, & \text{if } r_p^{\max} > r_p^q > r_p^{\min} \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

$r_p^{\min}$ is the minimum acceptable resolution in terms of pixel height at which pth block should be viewed, $r_p^{\max}$ is the maximum height of the target in pixels of c_i's image plane, and r_p^q is the resolution at which block p would be imaged for the proposed action a^{c_i}. The conditions under which this value function increases are when area covered with an acceptable resolution increases or the area can be viewed at a higher resolution.

Equation (6) implies that only cameras who are neighbors of c_i in communication graph can affect this utility value. By defining the local utility in such a way, the change in the value of the local utility, denoted by Δu_{area}, and global utility ΔU_{area}, by a change in the PTZ parameters of camera in c_i from a_i to b_i (where $a_i, b_i \in A_i$) is related by

$$\begin{aligned} \Delta u_{\text{area}} &= u_{\text{area}}\left(b_i, \mathbf{a}_{-i}^{c_i}\right) - u_{\text{area}}\left(a_i, \mathbf{a}_{-i}^{c_i}\right) \\ &= \sum_{l \in B^i} \left(1 - \prod_{j \in C_i} (1 - \beta_l^j)\right)\Bigg|_{b_i} \\ &\quad \sum_{l \in B^i} \left(1 - \prod_{j \in C_i} (1 - \beta_l^j)\right)\Bigg|_{a_i} \\ &= U_{\text{area}}(b_i, \mathbf{a}_{-i}) - U_{\text{area}}(a_i, \mathbf{a}_{-i}) = \Delta U_{\text{area}}. \end{aligned} \tag{8}$$

This is true because B^i contains all blocks that can be affected by a change in a_i in U_{area} and all the cameras that can affect the value of blocks in B^i are in the set $C_i \subseteq C$.

2) Local High Resolution Imaging: The local value function for high resolution imaging of the tracked objects can then be written as

$$u_{\text{image}}(\mathbf{a}^{c_i}) = \sum_{j \in T^i} \left(1 - \prod_{k \in C_i} (1 - \zeta_j^k)\right) s_j \tag{9}$$

where T^i is the target being viewed by the cameras in the set C_i. ζ_j^k is defined as

$$\zeta_j^k = \begin{cases} 1 - e^{-\lambda_2 \frac{r_j^k}{r_j^{\max}}}, & \text{if } r_j^{\max} > r_j^k > r_j^{\min} \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

where r_j^k is the resolution weighted by the probability $p_{j,l}$ of a target j is at block l, which is computed using $\hat{\mathbf{p}}_j^w(k)^+$ and

$\hat{\mathbf{P}}_j^w(k)^+$ and can be written as

$$r_j^k = \begin{cases} \sum_{b_l \in B_j} \left(p_{j,l} r_l^k\right), & \text{if } \forall b_l \in B_j, \quad r_l^k > 0 \\ 0, & \text{otherwise.} \end{cases} \tag{11}$$

The scaling parameter s_j in (9) is an importance factor to weight the value of getting high resolution images of target j by changing it dynamically over time. We populate this variable using a function monotonically increasing in time, initialized to a low value when a new track j is added to the system. Once a high resolution image of the track has been acquired $s_j = 0$ for the remaining lifetime of the track.

B. Communication and Vision Graphs

The change in the value of the local and global utility for high resolution imaging defined in (9), denoted by Δu_{image} and ΔU_{image}, respectively. The relationship between Δu_{image} and ΔU_{image} is given by

$$\begin{aligned} \Delta u_{\text{image}} &= u_{\text{image}}\left(b_i, \mathbf{a}_{-i}^{c_i}\right) - u_{\text{image}}\left(a_i, \mathbf{a}_{-i}^{c_i}\right) \\ &= \sum_{j \in T^{c_i}} \left(1 - \prod_{i \in C_i} (1 - \beta_l^i)\right)\Bigg|_{b_i} \\ &\quad - \sum_{j \in T^{c_i}} \left(1 - \prod_{i \in C_i} (1 - \beta_l^i)\right)\Bigg|_{a_i} \\ &= U_{\text{image}}(b_i, \mathbf{a}_{-i}) - U_{\text{image}}(a_i, \mathbf{a}_{-i}) = \Delta U_{\text{image}}. \end{aligned} \tag{12}$$

This is true because only targets who have a probability of being in B^i can be affected by a change in a_i, and the only cameras that can affect blocks in B^i are the neighboring cameras C_i.

1) Local Utility Function: Given the new local value functions for area coverage and high resolution imaging defined in (6) and (9). The utility function representing the local value of the system can be represented as

$$u_{\text{local}}(\mathbf{a}^{c_i}) = u_{\text{area}}(\mathbf{a}^{c_i}) + u_{\text{image}}(\mathbf{a}^{c_i}) \tag{13}$$

and the change in $\Delta u_{\text{local}}^{c_i}$, global utility ΔU_{global}, and the PTZ parameters of camera in c_i from a_i to b_i (where $a_i, b_i \in A_i$) is related by

$$\begin{aligned} \Delta u_{\text{local}}^{c_i} &= u_{\text{local}}\left(b_i, \mathbf{a}_{-i}^{c_i}\right) - u_{\text{local}}\left(a_i, \mathbf{a}_{-i}^{c_i}\right) \\ &= \Delta U_{\text{global}}. \end{aligned} \tag{14}$$

Since a change in the local utility $\Delta u_{\text{local}}^{c_i}$ is equivalent to the change in the global utility ΔU_{global} for any a_i. We can increase the global utility by increasing any camera's local utility.

C. Distributed Optimization Algorithm

The alignment between the local utility and the global utility is necessary to ensure that local decisions are coordinated with the global objective of the system. Given the local utility defined in (13) and its alignment to the global utility of the camera network, we modify the algorithm run at each camera i so that more than one camera can change PTZ parameters at each time step.

Step 1: c_i computes the best PTZ setting

$$\max_{a_i \in A_i} \left(\Delta u_{\text{local}}^{c_i}\right) \tag{15}$$

for camera i and sends the proposed setting a_i and utility $\max_{a_i \in A_i}(\Delta u_{\text{local}}^{c_i})$ to all neighboring cameras.

Step 2: Receive the proposed setting a_i' and utility $\max_{a_i' \in A_i'}(\Delta u_{\text{local}}^{c_i'})$ from each other camera in the network.

Step 3: If c_i can provide the greatest improvement

$$\max_{c_i' \in C_i} \left(\max_{a_i' \in A_i'} \left(\Delta u_{\text{local}}^{c_i'}\right)\right) = \max_{a_i \in A_i} \left(\Delta u_{\text{local}}^{c_i}\right) \tag{16}$$

then execute the proposed settings. In the case, multiple cameras have the same improvement any tie breaker such that $g(c_i, c_i') = g(c_i', c_i)$ can be used.

Step 4: Send current settings to all neighboring cameras.

Step 5: Receive current settings from all neighboring cameras and update $\mathbf{a}^{c_i}$.

D. Deterministic Equilibrium

We now analyze the convergence characteristics of the distributed optimization algorithm addressed in Section III-C and show that the cameras can arrive at an NE with one or more cameras deciding to change parameters simultaneously.

Definition 3: A path in the parameter space S of the camera network is a sequence of camera network parameters $(\mathbf{a}^1, \mathbf{a}^2, \ldots, \mathbf{a}^{N_S})$, such that each of the two consecutive camera network states differs in at least one camera PTZ setting.

Definition 4: An improvement path is a path in which $U_{\text{global}}(\mathbf{a}^k) < U_{\text{global}}(\mathbf{a}^{k+1})$, where $\mathbf{a}^k$ and $\mathbf{a}^{k+1}$ differ in at least one camera state, for all $k = 1, 2, \ldots$.

Definition 5: The camera network is at an NE if $\forall a_{-i} \in A_{-i}$ and $\forall a_i, b_i \in A_i$, $\Delta U_{\text{global}} = U_{\text{global}}(b_i, \mathbf{a}_{-i}) - U_{\text{global}}(a_i, \mathbf{a}_{-i}) \leq 0$ and $\Delta u_{\text{local}}^{c_i} = u_{\text{local}}(b_i, \mathbf{a}_{-i}^{c_i}) - u_{\text{local}}(a_i, \mathbf{a}_{-i}^{c_i}) \leq 0$.

Claim 1: Given a fixed environment and an arbitrary set of initial camera settings, the network of cameras will improve upon the global utility U_{global} until it reaches an NE.

Proof: As there are a finite number of camera network configurations, it is easy to see that every improvement path is finite and arrives at an NE. If the camera network is not at an NE, then there is a set of cameras such that $\max_{a_i \in A_i}(\Delta u_{\text{local}}^{c_i}) > 0$. After applying the condition in (16), the set of cameras that decide to move is $C_{\text{move}} \subseteq C$ resulting in the camera network state $\mathbf{a}_{\text{move}}$. Also, $\forall c_i', c_i'' \in C_{\text{move}}$, $c_i' \notin C_i''$ and $c_i'' \notin C_i'$. This means that the contribution of c_i' to the global utility is independent of the contribution of c_i''. The resulting change in the global value can then be written as

$$\Delta U_{\text{global}}|_{\mathbf{a}_{\text{move}}} = \sum_{i \in C_{\text{move}}} \left(\Delta u_{\text{local}}^{c_i}|_{\mathbf{a}_{\text{move}}}\right). \tag{17}$$

Since $\Delta u_{\text{local}}^{c_i}|_{\mathbf{a}_{\text{move}}} > 0, \forall i \in C_{\text{move}}$, then $\Delta U_{\text{global}}|_{\mathbf{a}_{\text{move}}} > 0$. Thus, by Definitions 4 and 5, the camera network is either at an NE or moves along an improvement path. ■

IV. Experimental Analysis and Results

In this section, we will explain the details of our experimental setup and our results. The experiments were performed on a wireless PTZ camera network consisting of Axis 215 PTZ-E cameras over a 20 m × 30 m region. Interactions with cameras in the wireless network are accomplished using the provided VAPIX API. This HTTP based API results in a significant lag in both the video processing and command execution. In our physical system, we noticed a delay ranging from 80 to 200 ms for each message to arrive over TCP. This limits the rate at which we can change the parameters of the camera network as there is a significant delay in the video response after sending a command.

The detection method we used in our experiments was very simple in design. First, the area was uniformly divided into a number of overlapping blocks, each 1 m × 1 m × 1.8 m. The frames from each camera are processed using a motion subtraction [32] algorithm to generate a motion image. The blocks in the camera operational region are then mapped to the image plane using the homography.

In this implementation, we make no assumptions as to the number of people present in each detection or track. Two people walking close together may result in a single detection with a covariance encompassing both. This means that if two people stay close together for the duration, they are in the region and they will be tracked as a group rather than as two individual tracks. Since we do not explicitly count the number of people in each track, a few special events, such as track split and track merge, can occur.

Split Track: When two or more people who are walking together start drifting apart, the detector will return multiple detections that associate to their existing track. The existing track is removed and one new track for each detection is created.

Merge Track: When two or more people start walking ever closer to each other, the detector will eventually return a single detection for all of the people. This event occurs when one detection associates to more than one track. In such cases, the existing tracks are removed and a new track is created for the group.

One thing to note here is that as people form groups or split apart, there is a short time period where multiple track split and track merge events may occur rapidly until they are close enough to form a group or far enough apart to be tracked separately. The scaling parameter s_j is tuned, such that high resolution images are not prioritized until after a track has existed for a period of time so as to avoid capturing multiple high resolution images during merge and split events.

Parameter Selection: There are many ways to speed up the search for the PTZ setting that maximizes the local utility, and the easiest approach is to increase the quantization of the parameter space. For our experiments, the available pan settings were quantized into 5° increments and bounded based on the position of the camera and the area under surveillance. The tilt settings were quantized into 1° increments and were restricted to a maximum tilt of 30°. Each camera also was restricted to three different zoom settings. Some additional techniques, such as a lookup table, were also used to improve the computation speed, such that each camera can detect, track, and decide on parameters within a 33-ms time window on a 2.66-GHz Intel Core i560m. The communication overhead of our physical setup reduces the rate at which our cameras change their parameters.

A. Experimental Results

We ran a number of different test scenarios through our system. A video of all the scenes is available at *http://www.ee.ucr.edu/ ∼amitrc/CameraNetworks.php*. Here, we will explain two of them (scenes 4 and 5 of the video) that will demonstrate all possible cases, such as track splitting, merging, and individual tracking. The objective is to cover the entire area while opportunistically acquiring high resolution shots of people in the area.

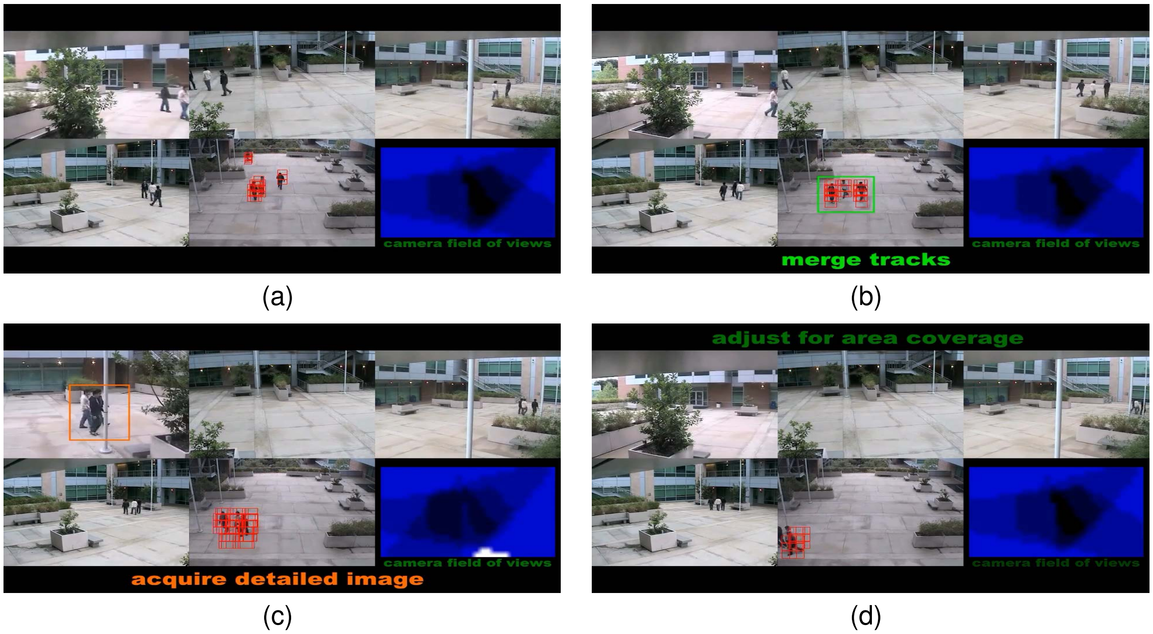
First Scenario: In this scenario, two pairs of people enter the scene from the opposite sides of the courtyard. One pair splits up shortly upon entering the area, while the other pair stays together. Some time after the first group split, one of its members walks toward and joins together with other pair forming a group of three as they proceed to exit the area.

The system is expected to form a track and acquire a high resolution image for each of the initial pairs of people. After the first group splits, two new tracks are expected to be created and high resolution images of each should be taken. When the track of the second pair and one of the members of the first pair merge, the system is expected to create a new track and capture a corresponding high resolution image.

Fig. 3 shows the behavior of the camera network in response to the merging of the tracks. In Fig. 3(a), we can see three tracks, two people walking alone and a pair of people walking together. As two of the tracks start getting closer, it becomes more difficult for our basic detector to separate the pair of people from the person walking alone. This can be seen in Fig. 3(b) as the tracks merge to form a group of three people. A high resolution image is shown in Fig. 3(c) a short while after the formation of the new track.

Second Scenario: The final scenario is the most complicated and shows how the system behaves when it does not have the time to complete all the desired tasks. In this scenario, four people enter the courtyard from each of the four corners within seconds of each other. They each walk toward the opposite end of the courtyard resulting in the four of them meeting in the center of the courtyard before reaching their final destinations.

As the paths of the people in the scene start intersecting [shown in Fig. 4(a)], it becomes more difficult to segment the detections of the individuals and results in the merging of all four people into a new single track, as shown in Fig. 4(b). Since this is a large group covering a significant portion of the area, very little area is needed to be sacrificed to take a high resolution image of the entire group resulting in the image shown in Fig. 4(c). During the time period where the four tracks are becoming merged into one, many short lived tracks are created as the detector begins to have difficult separating individual tracks. If the targets stay in a position where the



(a) (b) (c) (d)

Fig. 3. Scene involving a pair of people and two individual people. The images show the result when two of the tracks merge into a new group. (a) Two individuals and a group of two people are distinguishable by the detections, shown in red boxes, in the bottom middle camera. (b) Detections of the group of two and an individual in the scene can no longer be separated and a new group is formed. (c) High resolution image is acquired by the top-left camera. (d) Cameras reconfigure the cover the area.

detector cannot consistently separate or group them together, a long-term track will not exist and no high resolution images will be acquired. As the scene progresses, the people start to separate as can be seen in Fig. 4(c). A short while after the split, distinct tracks for each person can again be shown in Fig. 4(d). A sequence of high resolution images for each of the targets are then taken, as shown in Fig. 4(c)–(e). It is important to note that the third high resolution image is only taken after the other two zoomed-in cameras have returned to covering a significant portion of the area. As more cameras zoom in for high resolution images, the amount of area covered by overlapping FOVs decreases. Due to the tradeoff between the value gained from area coverage and high resolution imaging, the less cameras responsible for covering the area, the less likely a camera will zoom in for a detailed image.

B. Analysis of Active Selection

We analyze the active vision scenario (high resolution shots) in terms of number of pixels. We determine an average and maximum number of pixels with and without active selection of cameras. The number of pixels is calculated from every frame within a specific time span for both with and without zoom and averaged over total frames to find the average number of pixels. In Table II, an average and maximum number of pixels of whole target and face region of the targets are provided to demonstrate the effect of active control in terms of resolution. Since the operations in the aforementioned scenarios are similar, we choose four targets from the final scenario to compute the number of pixels. In final scenario, all targets have been zoomed in twice—before track merging and after track splitting. Only target 1 has been zoomed in once—before track merging and after track splitting, it is lost. High resolution shot acquired through active selection strategy gives us detailed information about targets.

TABLE II
QUANTITATIVE ANALYSIS OF ACTIVE CONTROL

Targets	**Avg. Nb of Pixels (with No control)**	**Avg. Nb of Pixels (with control)**	**Max. Nb of Pixels (with No control)**	**Max. Nb of Pixels (with control)**
1. Face	171.70	1058.27	231.00	4228.20
Whole Body	515.11	3386.48	693.00	12896.00
2. Face	176.81	735.15	480.00	4318.94
Whole Body	530.44	2278.96	1440.00	13000.00
3. Face	317.67	852.57	480.00	4201.32
Whole Body	953.01	2728.21	1440.00	12688.00
4. Face	321.00	785.21	783.33	4819.74
Whole Body	963.00	2434.14	2350.00	14652.00

1) Network Delay: In the second scenario as shown in Fig. 4(i), we can see that a high resolution image of one of the targets was not acquired before he exited the area after the group split. This is because the time span from when the individual track was created until the time he left the area was too short for any camera to gain utility from capturing a high resolution image. If the delay of the network was very small or almost instantaneous, it would have been possible to capture the high resolution shot of the target.

2) Communication Cost: In the centralized scheme, data need to be present at a central server. As the number of cameras increases, the communication cost becomes high. There are many distributed frameworks, some are consensus based and some are not. The nonconsensus-based distributed network presented in [16] allows only one camera to change their PTZ setting at each iteration time. When we scale the size of the camera network, this can quickly become a problem. To overcome this, we design a distributed system in a way that only camera nodes that are part of the vision graph will exchange information with each other, since they sense the same target. The number of such cameras looking at a

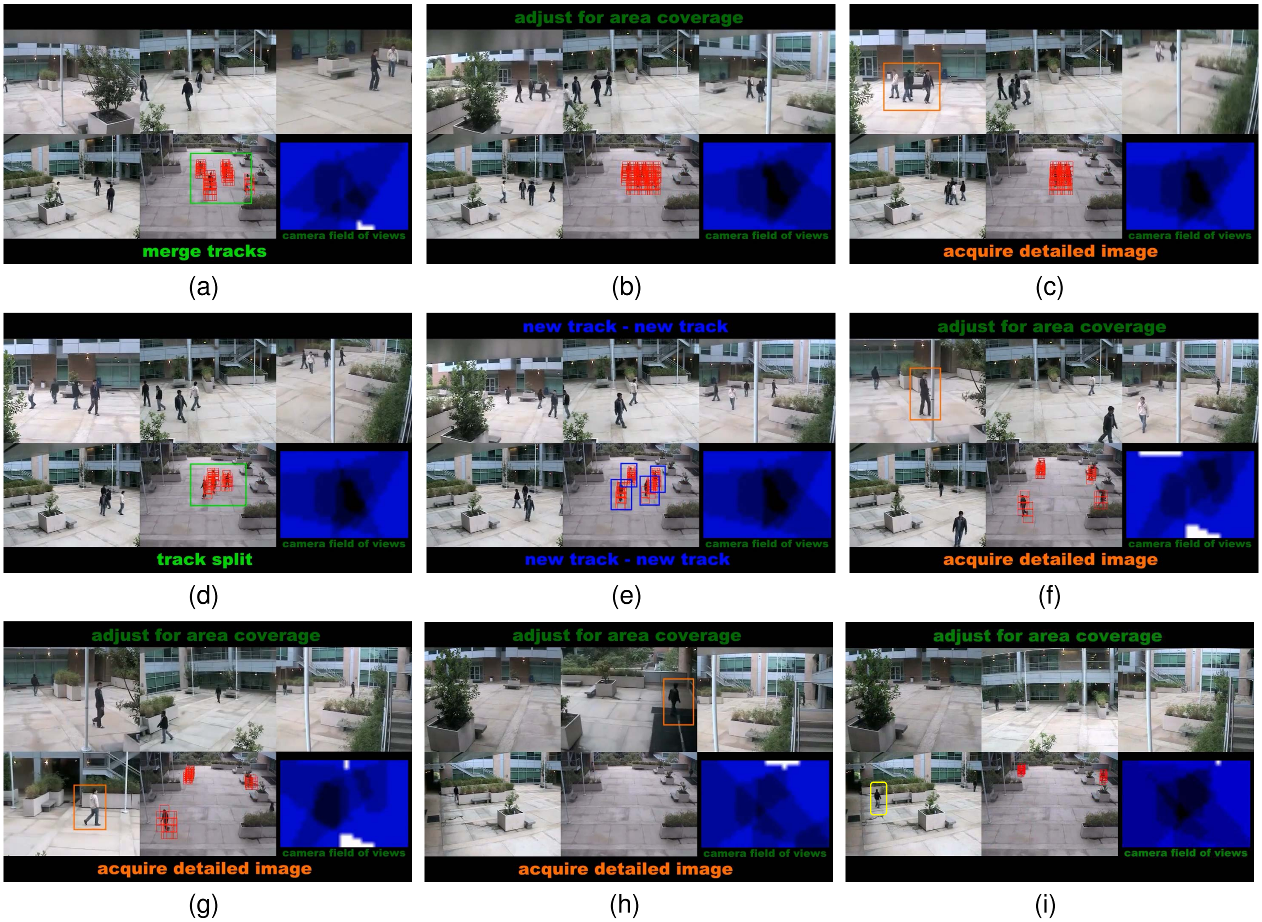


Fig. 4. Sequence of images taken in response to merging and splitting tracks. (a) As the targets move closer together, their tracks begin to merge into a single track. (b) Four people have grouped together to create a new track. (c) High resolution image of the entire group is taken by the upper left camera. (d) Targets are beginning to split apart. (d) New track is created for each individual as there is enough separation to generate stable tracks. (f)–(h) High resolution images are acquired for three of the individual tracks. (i) Fourth target is leaving the surveillance area before the network could take a high resolution shot (yellow box).

TABLE III

DATA TRANSFER RATE FOR DISTRIBUTED (WITH DIFFERENT CLIQUE SIZES) AND CENTRALIZED SCHEME

Distributed Scheme		Centralized Scheme
Clique-2	Clique-3	
5.750 MB/s	8.775 MB/s	15.125 MB/s

particular target will usually be small in most applications. Thus, the amount of data that is exchanged between these cameras is much lower than a centralized case, since central server needs to access all the information from all the cameras in the network. For example, if there are a total of N_c cameras in the network and N_v in a particular clique of a vision graph (i.e., cameras that sense the same target and thus need to exchange information about that target), the relative amount of data exchanged in the distributed case is N_v/N_c of the centralized case (approximately). For typical values of N_v and N_c, we can see that the distributed case provides significant reduction in communication cost. In Table III, the data transfer rate for distributed and central network has been provided. Here, $N_c = 5$ and $N_v = \{2, 3\}$ are used in vision graph.

V. CONCLUSION

We described an active camera parameter selection method for capturing high resolution images of targets, when certain events occur, while maintaining coverage of a large area using a camera network. A key contribution of this paper is showing how the parameter selection can be distributed in a network so that multiple cameras can change parameters simultaneously while prior approaches allowed only one camera to update its parameters at a time. Future work should consider codesign methodologies whereby the effect of communication limitations or processing capabilities is modeled in the camera control objective functions.

REFERENCES

[1] A. K. Roy-Chowdhury and B. Song, “Camera networks: The acquisition and analysis of videos over wide areas,” *Synth. Lectures Comput. Vis.*, vol. 3, no. 1, pp. 1–133, 2012.

[2] J. Zhao, S.-C. Cheung, and T. Nguyen, “Optimal camera network configurations for visual tagging,” *IEEE J. Sel. Topics Signal Process.*, vol. 2, no. 4, pp. 464–479, Aug. 2008.

[3] Z. Tu and P. Bhattacharya, “Game-theoretic surveillance over arbitrary floor plan using a video camera network,” *Signal, Image Video Process.*, vol. 7, no. 4, pp. 705–721, Jul. 2013. [Online]. Available: http://dx.doi.org/10.1007/s11760-013-0484-8

[4] U. M. Erdem and S. Sclaroff, “Automated camera layout to satisfy task-specific and floor plan-specific coverage requirements,” *Comput. Vis. Image Understand.*, vol. 103, no. 3, pp. 156–169, 2006.

[5] R. Tron and R. Vidal, “Distributed computer vision algorithms,” *IEEE Signal Process. Mag.*, vol. 28, no. 3, pp. 32–45, May 2011.

[6] M. Taj and A. Cavallaro, “Distributed and decentralized multicamera tracking,” *IEEE Signal Process. Mag.*, vol. 28, no. 3, pp. 46–58, May 2011.

[7] B. Dieber, C. Micheloni, and B. Rinner, “Resource-aware coverage and task assignment in visual sensor networks,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 10, pp. 1424–1437, Oct. 2011.

[8] A. D. Bagdanov, A. del Bimbo, and F. Pernici, “Acquisition of high-resolution images through on-line saccade sequence planning,” in *Proc. 3rd ACM Int. Workshop Video Surveill. Sensor Netw.*, 2005, pp. 121–130.

[9] A. Blake and A. Yuille, Eds., *Active Vision*. Cambridge, MA, USA: MIT Press, 1992.

[10] F. Z. Qureshi and D. Terzopoulos, “Planning ahead for PTZ camera assignment and handoff,” in *Proc. IEEE/ACM Int. Conf. Distrib. Smart Cameras*, Como, Italy, Aug./Sep. 2009, pp. 1–8.

[11] A. Mittal and L. S. Davis, “A general method for sensor planning in multi-sensor systems: Extension to random occlusion,” *Int. J. Comput. Vis.*, vol. 76, no. 1, pp. 31–52, 2008.

[12] E. Sommerlade and I. Reid, “Information-theoretic active scene exploration,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–7.

[13] C. Piciarelli, C. Micheloni, and G. L. Foresti, “PTZ camera network reconfiguration,” in *Proc. IEEE/ACM Int. Conf. Distrib. Smart Cameras*, Como, Italy, Aug./Sep. 2009, pp. 1–8.

[14] C. Micheloni, B. Rinner, and G. L. Foresti, “Video analysis in pan-tilt-zoom camera networks,” *IEEE Signal Process. Mag.*, vol. 27, no. 5, pp. 78–90, Sep. 2010.

[15] B. Song, C. Ding, A. T. Kamal, J. A. Farrell, and A. K. Roy-Chowdhury, “Distributed wide area scene analysis in reconfigurable camera networks,” *IEEE Signal Process. Mag.*, vol. 28, no. 3, pp. 20–31, May 2011.

[16] C. Ding, B. Song, A. Morye, J. A. Farrell, and A. K. Roy-Chowdhury, “Collaborative sensing in a distributed PTZ camera network,” *IEEE Trans. Image Process.*, vol. 21, no. 7, pp. 3282–3295, Jul. 2012.

[17] A. A. Morye, C. Ding, B. Song, A. K. Roy-Chowdhury, and J. A. Farrell, “Optimized imaging and target tracking within a distributed camera network,” in *Proc. Amer. Control Conf.*, Jun./Jul. 2011, pp. 474–480.

[18] N. Li and J. R. Marden, “Designing games for distributed optimization,” in *Proc. 50th IEEE Conf. Decision Control Eur. Control Conf.*, Dec. 2011, pp. 2434–2440, doi: 10.1109/CDC.2011.6161053.

[19] S. Pellegrini, A. Ess, and L. Van Gool, “Improving data association by joint modeling of pedestrian trajectories and groupings,” in *Computer Vision—ECCV*, K. Daniilidis, P. Maragos, and N. Paragios, Eds. Berlin, Germany: Springer, 2010, pp. 452–465.

[20] Z. Qin and C. R. Shelton, “Improving multi-target tracking via social grouping,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2012, pp. 1972–1978.

[21] A. Chakraborty, A. Das, and A. K. Roy-Chowdhury, “Network consistent data association,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 9, pp. 1859–1871, Sep. 2016.

[22] X. Chen, Z. Qin, L. An, and B. Bhanu, “An online learned elementary grouping model for multi-target tracking,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 1242–1249.

[23] L. Bazzani, M. Cristani, and V. Murino, “Decentralized particle filter for joint individual-group tracking,” in *Proc. CVPR*, Jun. 2012, pp. 1886–1893.

[24] W.-S. Zheng, S. Gong, and T. Xiang, “Associating groups of people,” in *Proc. BMVC*, vol. 2. 2009, pp. 23.1–23.11.

[25] Y. Cai, V. Takala, and M. Pietikäinen, “Matching groups of people by covariance descriptor,” in *Proc. ICPR*, Aug. 2010, pp. 2744–2747.

[26] R. Vezzani, D. Baltieri, and R. Cucchiara, “People reidentification in surveillance and forensics: A survey,” *ACM Comput. Surv.*, vol. 46, no. 2, pp. 29:1–29:37, Nov. 2013.

[27] A. Yilmaz, O. Javed, and M. Shah, “Object tracking: A survey,” *ACM Comput. Surv.*, vol. 38, no. 4, Dec. 2006, Art. no. 13. [Online]. Available: http://doi.acm.org/10.1145/1177352.1177355

[28] T. Ueshiba and F. Tomita, “Plane-based calibration algorithm for multi-camera systems via factorization of homography matrices,” in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2003, pp. 966–973.

[29] A. T. Kamal, J. H. Bappy, J. A. Farrell, and A. K. Roy-Chowdhury, “Distributed multi-target tracking and data association in vision networks,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 7, pp. 1397–1410, Jul. 2016.

[30] A. T. Kamal, J. A. Farrell, and A. K. Roy-Chowdhury, “Information weighted consensus filters and their application in distributed camera networks,” *IEEE Trans. Autom. Control*, vol. 58, no. 12, pp. 3112–3125, Dec. 2013.

[31] A. T. Kamal, J. A. Farrell, and A. K. Roy-Chowdhury, “Information consensus for distributed multi-target tracking,” in *Proc. CVPR*, Jun. 2013, pp. 2403–2410.

[32] Z. Zivkovic, “Improved adaptive Gaussian mixture model for background subtraction,” in *Proc. 17th Int. Conf. Pattern Recognit. (ICPR)*, vol. 2. 2004, pp. 28–31.

Chong Ding (S’10) received the B.S. and Ph.D. degrees in computer science from the University of California at Riverside, Riverside, CA, USA, in 2008 and 2013, respectively.

He is currently a Research Staff Member with the Information and System Sciences Laboratory, HRL Laboratories, LLC, Malibu, CA, USA. He has authored several technical publications on intelligent camera networks. His current research interests include autonomous vehicles, mobile ad-hoc networks, network security, and distributed and real-time systems.

Jawadul H. Bappy (S’16) received the B.S. degree in electrical and electronic engineering from the Bangladesh University of Engineering and Technology, Dhaka, Bangladesh, in 2012. He is currently pursuing the Ph.D. degree in electrical and computer engineering with the University of California at Riverside, Riverside, CA, USA.

His current research interests include wide area scene analysis, scene understanding, object recognition, and machine learning.

Jay A. Farrell received the B.S. degrees in physics and electrical engineering from Iowa State University, Ames, IA, USA, and the M.S. and Ph.D. degrees in electrical engineering from the University of Notre Dame, Notre Dame, IN, USA.

He is currently a Professor and two-time Chair of the Department of Electrical Engineering with the University of California at Riverside, Riverside, CA, USA. He has authored over 200 technical publications and the book entitled *Aided Navigation: GPS With High Rate Sensors* (McGraw-Hill, 2008). He has co-authored the books entitled *The Global Positioning System and Inertial Navigation* (McGraw-Hill, 1998) and *Adaptive Approximation Based Control: Unifying Neural, Fuzzy and Traditional Adaptive Approximation Approaches* (John Wiley, 2006). His current research interests include applied research related to estimation, planning, and control of intelligent autonomous agents.

Dr. Farrell is a fellow of the American Association for the Advancement of Science and a Distinguished Member of the IEEE Control Systems Society (CSS). He has served as the Vice President Finance and Vice President of Technical Activities of the IEEE CSS, and served as the General Chair of the IEEE CDC in 2012 and the President in 2014.

Amit K. Roy-Chowdhury (SM’98) received the bachelor’s degree in electrical engineering from Jadavpur University, Kolkata, India, the master’s degree in systems science and automation from the Indian Institute of Science, Bangalore, India, and the Ph.D. degree in electrical engineering from the University of Maryland, College Park, MD, USA.

He is currently a Professor of Electrical Engineering and a Cooperating Faculty Member with the Department of Computer Science, University of California at Riverside, Riverside, CA, USA. He has co-authored two books entitled *Camera Networks: The Acquisition and Analysis of Videos Over Wide Areas* and *Recognition of Humans and Their Activities Using Video*. He is the Editor of the book entitled *Distributed Video Sensor Networks*. His current research interests include the areas of image processing and analysis, computer vision, video communications and statistical methods for signal analysis, intelligent camera networks, wide-area scene analysis, motion analysis in video, activity recognition and search, video-based biometrics (face and gait), biological video analysis, and distributed video compression.

Dr. Roy-Chowdhury has been on the organizing and program committees of multiple computer vision and image processing conferences and is serving on the editorial boards of multiple journals.