2011 American Control Conference
on O'Farrell Street, San Francisco, CA, USA
June 29 - July 01, 2011

Optimized Imaging and Target Tracking within a Distributed Camera Network*

A. A. Morye, C. Ding, B. Song, A. Roy-Chowdhury, J. A. Farrell
Department of Electrical Engineering
University of California, Riverside

Abstract— **This article considers the problem of using a network of N_C dynamic pan, tilt, zoom cameras, each mounted at known and fixed locations, to track and obtain high resolution imagery for $N_T(t)$ mobile targets each maneuvering within a confined space. The number of targets is time-varying, the targets are free to maneuver, the targets may enter or leave the region under surveillance so that $N_T(t)$ is time-varying and may exceed N_C.**

Tracking **a target is defined as estimating the position of the target with horizontal uncertainty less that a specified threshold $\bar{P}$.** ***Imaging*** **a target is defined as obtaining an image with vertical resolution exceeding $\bar{r}$. The problem is to organize the pan, tilt, and zoom parameters of the network of cameras at each sampling instant such that the tracking specification for all targets and the imaging specification for specific targets at times of opportunity is achieved. This problem could be addressed by centralized or decentralized methods. In this article, we are focused on distributed control of the camera network.**

We develop a distributed optimization solution, where we consider each camera to be an individual decision making agent. The solution involves formulation of the approach, design of a value function, and design of a probability-based camera ordering mechanism to aid convergence of the distributed network solution towards an optimal solution. Our approach is developed within a Bayesian approach to appropriately trading-off value V (target tracking accuracy and target resolution) versus risk (probability of losing track of a target). This article presents the theoretical solution along with simulation results. Implementation on a camera network are in progress.

I. INTRODUCTION

Networks of video cameras are being installed for a variety of applications, such as surveillance and security, environmental monitoring, and disaster response. Existing camera networks consist mostly of fixed cameras covering large areas. This results in situations where some targets are often not covered at the desired resolutions or viewpoints, making the analysis of the video difficult, while some cameras are imaging space that is devoid of interesting entities. Networks of actively controlled pan, tilt, zoom cameras could allow for maximal utilization of the imaging resources by allowing the cameras to differentially focus on multiple regions of interest through dynamic camera parameter selection. Such a setup will thus provide greater flexibility while requiring less hardware.

A prototypical application is a security screening checkpoint at the entrance to a building. Over the course of each day a high volume of people flow through the room. The number of cameras is fixed while the number of persons in the room is time varying. At any given time, the objective of the camera network is to maintain tracking (i.e., state estimation) for all persons in the room and to capture high resolution images for certain persons in the room.

In this paper, we focus on the problem of controlling the cameras in a wide-area active camera network so as to maximize multi-target tracking performance. In order to achieve this, it is necessary to assign the camera parameters, dynamically so as to obtain high fidelity target imagery and tracking. This means that based on the tracks and tracking error estimates, we need to control the cameras so as to minimize the tracking error and get imagery at the desired resolutions and poses. It is also desirable in many applications for the tracking and control mechanisms to be distributed due to bandwidth and security constraints. This would require each camera to act as an autonomous agent and cooperatively track targets and decide actions.

II. PROBLEM DESCRIPTION

The overall goal of this paper is to develop distributed camera control for optimal tracking in wide-area environments. Our problem domain envisions a number of cameras N_C placed in and around the region under surveillance and a time-varying number of targets $N_T(t)$. These cameras have known fixed locations with dynamic pan ρ, tilt τ, and zoom ζ parameters. In a decentralized framework, the cameras cooperate such that each camera selects its dynamic parameters (ρ, τ, ζ) to optimize a global value function cooperatively. The target locations vary with time in a manner that is not known *a priori* to the cameras; therefore, the state of each target must be estimated from the camera imagery.

Targets are not directly assigned to cameras. Instead, the i-th camera selects its parameters $\mathbf{a}_i = (\rho_i, \tau_i, \zeta_i)$, which results in a field-of-view (FOV) for the resulting image. That image may contain multiple targets and each target may be imaged by multiple cameras.

At the time that each camera selects its parameters for the image scheduled to occur at the future time t_{k+1}, the target locations at t_{k+1} are unknown. Based on the last set of imagery from time t_k, the target state estimation process provides a prior mean $\hat{\mathbf{x}}^j(k+1)^-$ and covariance matrix $\mathbf{P}^j(k+1)^-$ for all targets (i.e., $j = 1, \ldots, N_T$). Due to uncertainty in the target state $\mathbf{x}^j(k+1)$ there is a tradeoff

*Research supported by ONR under award no.: N000140910666 for the project titled 'Distributed Dynamic Scene Analysis in a Self-Configuring Multimodal Sensor Network'.

in the camera parameter selection between tracking gain and coverage risk. Therefore, we develop our approach within a Bayesian framework.

A. Overall Problem

Consider the time interval $t \in (t_k, t_{k+1})$ where t_k is the time of the last set of images and t_{k+1} is the time scheduled for the next set of images. During this time interval several processes must be accomplished, see Fig. 1. Each of the cameras in our network has its own embedded target detection module, an Extended Kalman-Consensus tracker [1], [2] that provides a distributed consensus estimate on the state of each target, and finally a distributed camera parameter selection mechanism. Fig. 2 depicts the series of temporal events. Below the timeline in Fig. 2 variables are listed at the time that they are available. Notation as described below is summarized in Table I.

The first process is target detection. The target detection module in each camera takes its raw image and returns the image plane positions of each target recognized in the image. Communication between cameras is allowed to enhance the processes of feature detection and association for target recognition [3]. In Fig. 2 the time of completion of this process is denoted as t_β. At t_β, each camera has computed the pixel coordinate measurement of each recognized target within its FOV. Assuming that target j is within the FOV of camera i, this image frame measurement of the pixel location of target j by camera i valid at time t_k is denoted by ${}^i\mathbf{u}^j(k)$. This measurement is broadcast to neighboring cameras.

The second process is target state estimation. Using its own image plane position measurements and those received from the camera network, each camera implements a consensus state estimation algorithm [1], [2], [3], [4] to compute a posterior mean $\hat{\mathbf{x}}^j(k)^+$ and covariance matrix $\mathbf{P}^j(k)^+$ for all targets (i.e., $j = 1, \ldots, N_T$). Using the posterior information from t_k and the assumed target model, the prior mean $\hat{\mathbf{x}}^j(k+1)^-$ and covariance matrix $\mathbf{P}^j(k+1)^-$ for all targets is computed as an input to the camera parameter selection process. In Fig. 2 the time of availability of the prior information is indicated as t_δ.

The third process is selection of the camera parameters for the next image. This process is the main focus of the present article. In Fig. 2, the parameter selection process occurs for $t \in (t_\delta, t_\epsilon)$, leaving the interval $t \in (t_\epsilon, t_{k+1})$ for the cameras to achieve the commanded parameter settings.

TABLE I
NOTATION SUMMARY

Parameter	Variable
Pan, Tilt, Zoom	(ρ, τ, ζ)
Focal length	F
No. of Cameras, No. of Targets in region	N_C, N_T
i-th camera, j-th target	C_i, T^j
(ρ, τ, ζ) settings for C_i, all cameras except C_i	$\mathbf{a}_i$, $\mathbf{a}_{-i}$
(ρ, τ, ζ) settings for all cameras	$\mathbf{a}$
Tracking Utility, Imaging Utility for T^j	$U_T^j(\mathbf{a})$, $U_I^j(\mathbf{a})$
Global Utility	$U(\mathbf{a})$
Expected value of $U(\mathbf{a})$ over all targets	$V(\mathbf{a})$
Weight for importance of imagery of T^j	w^j
State vector for T^j	$\mathbf{x}^j$
State est., state est. covariance for T^j	$\hat{\mathbf{x}}^j$, $\mathbf{P}^j$
Fisher Information Matrix	$\mathbf{J}$
Measurement Vector, Measurement Covariance	$\mathbf{u}$, $\mathbf{C}$
Rotation Matrix from frame a to frame b	${}^b_a\mathbf{R}$
Entity b before, after new measurement	b^-, b^+
Entity b in global frame, frame defined by C_i	${}^gb$, ${}^ib$
Entity b at time-step t_k	$b(k)$
Entity b for target T^j	b^j

The camera parameter selection process is designed as a distributed optimization. Let $\mathbf{a}_i$, $\mathbf{a}_{-i}$, and $\mathbf{a}$, respectively, represent the vector of parameter settings for the i-th camera, all cameras other than the i-th camera, and all cameras. At the time that camera i is adjusting $\mathbf{a}_i$ the parameters in $\mathbf{a}_{-i}$ are held constant. Over the time interval $t \in (t_\delta, t_\epsilon)$, each camera will have various opportunities to adjust its parameter settings and communicate its revised settings to the network, such that the entire vector $\mathbf{a}$ converges towards the optimal settings for the upcoming image at t_{k+1}.

It must be noted that cameras take images at times $t_\kappa = \frac{\kappa T}{M}$, where κ is the image number and M is the number of frames the designer may choose to have between performing the parameter selection process. Thus, optimization occurs every $t_k = t_\kappa M$, and measurements and KF time propagation occur each t_κ. If desired, the designer can have $M = 1$.

The sequence of activities repeats in the time interval between any two images.

B. High Resolution Image Capture

In addition to target tracking we are interested in obtaining high-resolution imagery for certain targets. The importance of imagery for specific targets is indicated by weights $\{w^j\}_{j=1}^{N_T}$ in the utlity function. This weight can be made to change subject to scene analysis or if prior high-resolution

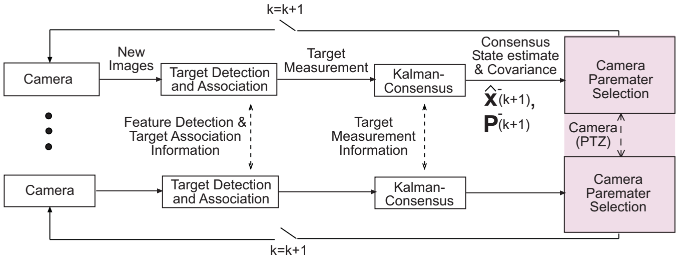
k=k+1
Camera
New Images
Target Detection and Association
Target Measurement
Kalman-Consensus
Consensus State estimate & Covariance
$\hat{\mathbf{x}}^-(k+1)$,
$\mathbf{P}^-(k+1)$
Camera Paremater Selection
Feature Detection & Target Association Information
Target Measurement Information
Camera (PTZ)
Camera
Target Detection and Association
Kalman-Consensus
Camera Paremater Selection
k=k+1

Fig. 1. Information exchange shown is only between neighboring cameras.

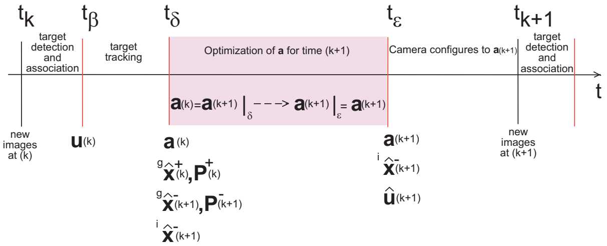
t_k
t_β
t_δ
t_ε
t_{k+1}
target detection and association
target tracking
Optimization of a for time (k+1)
Camera configures to a(k+1)
target detection and association
t
$\mathbf{a}(k)=\mathbf{a}(k+1)|_\delta$ - - -> $\mathbf{a}(k+1)|_\varepsilon = \mathbf{a}(k+1)$
new images at (k)
$\mathbf{u}(k)$
$\mathbf{a}(k)$
${}^g\hat{\mathbf{x}}^+(k), \mathbf{P}^+(k)$
${}^g\hat{\mathbf{x}}^-(k+1), \mathbf{P}^-(k+1)$
${}^i\hat{\mathbf{x}}^-(k+1)$
$\mathbf{a}(k+1)$
${}^i\hat{\mathbf{x}}^-(k+1)$
$\hat{\mathbf{u}}(k+1)$
new images at (k+1)

Fig. 2. Timeline of events between image sample times.

imagery of the target has been performed. Imagery from specific aspect angles may also be desirable and would be achieved using the assumption that the aspect angle is related to the direction of target motion. By including resolution specifications, aspect angle, and target importance in the utility function, we can further enhance the performance of the network, by making it possible to procure high resolution images of targets.

C. Related Work

The research presented in this paper is related to active vision [5], [6], [7]. Active vision in a camera network is a relatively unexplored area that involves cooperation and coordination between many cameras. There is a large amount of recent work dealing with networks of vision sensors. Some recent work has dealt with computing the statistical dependence between cameras, computing the camera network topology, tracking over unobserved areas of the network, and camera handoff [3], [8], [9], [10], [11], [12], [13]. However, there is little work that deals with distributed tracking and control in active camera networks. The most relevant papers on the topics of tracking and camera parameter selection are discussed below.

In [14], a distributed cluster-based Kalman filter was proposed as a target tracking approach. This method required a camera to aggregate all the measurements of a target to estimate its position before transmitting the result to a central base station. The approach in [3], used herein, considers a different network topology where each camera can only communicate with its neighboring cameras. Each camera has a consensus-based estimate of each target's state removing the need to aggregate measurements at a single cluster head.

A method for tracking targets in a network of PTZ cameras was proposed in [15]. The authors used a mixture of passive and active PTZ cameras to persistently track pedestrians in a virtual environment. This was achieved using a partially distributed, partially centralized hybrid approach. In our method we consider a completely distributed solution using consensus algorithms for tracking and a distributed optimization framework for camera parameter selection.

An interesting game-theoretic consensus based approach to the agent-target assignment problem was proposed in [16]. That article is not related to camera networks and addresses a different class of problems: targets are stationary at known locations, agents are mobile with known locations, one target is explicitly assigned to each agent.

A game-theoretic approach to camera control, limited to the area coverage problem was presented in [4]. The authors proposed a distributed tracking and control approach that requires the camera control and tracking to run independently and in parallel. The camera control used game theory to assign camera settings that provided coverage over regions of interest while maintaining a high resolution shot of a target. Concurrently, a Kalman-Consensus filter provided tracks of each target on the ground plane.

Our proposed method differs from this in that the camera control is aware of the state of the Kalman-Consensus filter and actively seeks to provide it with the best measurements. Furthermore our approach considers the estimate error covariance in addition to the estimated state of each target. This allows us to gauge the risk of failing to capture a feature when attempting high resolution shots. Our goal in this paper is to show that through active control of cameras we can minimize the tracking error of targets in a network of cameras.

III. System Model

The position of the i-th camera in the global frame is indicated by ${}^g\mathbf{p}_i$. In addition to the global frame, each camera defines a frame of reference. The position of T^j in the global frame would be indicated as ${}^g\mathbf{p}^j$ and in the frame of the i-th camera as ${}^i\mathbf{p}^j$. The time propagation models [17] for state estimation of T^j are stated below.

A. Time propagation models

The continuous-time state space model of target T^j is:

$$\dot{\mathbf{x}}^j(t) = \mathbf{F}\mathbf{x}^j(t) + \mathbf{G}\boldsymbol{\omega}^j(t) \qquad (1)$$

where, $\mathbf{x}^j = [{}^g\mathbf{p}^j; {}^g\mathbf{v}^j]$, where ${}^g\mathbf{p}^j$ and ${}^g\mathbf{v}^j$ are position and velocity, and $j = 1, \ldots, N_T$ is the target number. The process noise vector $\boldsymbol{\omega} \in \Re^3$ is zero mean Gaussian with power spectral density $\mathbf{Q}$.

The discrete-time equivalent model is:

$$\mathbf{x}^j(k+1) = \boldsymbol{\Phi}\mathbf{x}^j(k) + \boldsymbol{\gamma}(k) \qquad (2)$$

Here, $\boldsymbol{\Phi} = e^{\mathbf{F}T}$, $\boldsymbol{\gamma} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q_d})$, and $T = t_{k+1} - t_k$ is the sampling period. Thus, the state estimate and its error covariance matrix are propagated between sampling instants using:

$$\begin{aligned} \hat{\mathbf{x}}^j(k+1)^- &= \boldsymbol{\Phi}\hat{\mathbf{x}}^j(k)^+ && (3) \\ \mathbf{P}^j(k+1)^- &= \boldsymbol{\Phi}\mathbf{P}^j(k)^+\boldsymbol{\Phi}^\top + \mathbf{Q}_d && (4) \end{aligned}$$

B. Coordinate Transformations

Target T^j's position in the i-th camera frame is related to its position in the global frame by:

$$\begin{aligned} {}^g\mathbf{p}^j &= {}^g_i\mathbf{R}\,{}^i\mathbf{p}^j + {}^g\mathbf{p}_i && (5) \\ {}^i\mathbf{p}^j &= {}^i_g\mathbf{R}[{}^g\mathbf{p}^j - {}^g\mathbf{p}_i]. && (6) \end{aligned}$$

where ${}^i_g\mathbf{R}$ is a rotation matrix that is a function of the camera mounting angle, the pan angle, and the tilt angle.

C. Measurement Model

This section presents the nonlinear and linearized measurement models for target T^j when imaged by camera i. The linearization is performed relative to the targets estimated position ${}^g\hat{\mathbf{p}}^j$. In the remainder of this section, all measurement vectors are computed at t_k. The time argument and subscripts are dropped to simplify the notation where understanding of the material is not compromised.

We assume that positions ${}^g\hat{\mathbf{p}}^j$ and ${}^g\mathbf{p}_i$ are known, that the rotation matrix ${}^i_g\mathbf{R}(\rho_i, \tau_i)$ is a known function of the pan and the tilt angles, and that the focal length F_i is a known function of the zoom setting ζ_i.

Let the coordinates of target T^j in the i-th camera frame be ${}^i\mathbf{p}^j = \left[{}^ix^j, {}^iy^j, {}^iz^j\right]^\top$. Using the standard pinhole camera model with perspective projection [18], the projection of ${}^i\mathbf{p}^j$ onto the image plane of camera i is ${}^i\mathbf{u}^j = \left[\; F_i \frac{{}^ix^j}{{}^iz^j}, \;\; F_i \frac{{}^iy^j}{{}^iz^j}, \;\; F_i \;\right]^\top$. Thus, the image plane measurement ${}^i\mathbf{u}^j$ is:

$$ {}^i\mathbf{u}^j = \left[\begin{array}{c} F_i \frac{{}^ix^j}{{}^iz^j} \\ F_i \frac{{}^iy^j}{{}^iz^j} \end{array}\right] + {}^i\boldsymbol{\eta}^j \tag{7}$$

where the measurement noise ${}^i\boldsymbol{\eta}^j \sim \mathcal{N}(\mathbf{0}, \mathbf{C}_i^j)$ with $\mathbf{C}_i^j > \mathbf{0}$ and $\mathbf{C}_i^j \in \Re^{2\times2}$.

Given the estimated state and the camera model, the predicted estimate of the measurement is:

$$ {}^i\hat{\mathbf{u}}^j = \left[\begin{array}{c} F_i \frac{{}^i\hat{x}^j}{{}^i\hat{z}^j} \\ F_i \frac{{}^i\hat{y}^j}{{}^i\hat{z}^j} \end{array}\right]. \tag{8}$$

The measurement residual ${}^i\tilde{\mathbf{u}}^j$ is defined as:

$$ {}^i\tilde{\mathbf{u}}^j = {}^i\mathbf{u}^j - {}^i\hat{\mathbf{u}}^j. \tag{9}$$

D. Observation Matrix $\boldsymbol{H}_i^j$

Given ${}^g\mathbf{p}_i$, ${}^g\hat{\mathbf{p}}^j$, and ${}^i_g\mathbf{R}$, subsequent analysis will use the linearized relationship given by the first order Taylor series expansion of eqn. (7) around the estimated state. The linear relationship between the residual and the state error vector is:

$$ {}^i\mathbf{u}^j - {}^i\hat{\mathbf{u}}^j \approx \mathbf{H}_i^j({}^g\mathbf{p}^j -{}^g\hat{\mathbf{p}}^j) \tag{10}$$

where $\mathbf{H}_i^j = \left.\frac{\partial {}^i\mathbf{u}^j}{\partial {}^g\mathbf{p}^j}\right|_{{}^g\hat{\mathbf{p}}^j} \in \Re^{2\times3}$. Taking the partial derivatives as defined above, it is straightforward to show that:

$$ \mathbf{H}_i^j = \frac{F_i}{({}^i\hat{z}^j)^2}\left[\begin{array}{c} {}^g\mathbf{N}_1^{j^\top} \\ {}^g\mathbf{N}_2^{j^\top} \end{array}\right] \tag{11}$$

where,

$$ \begin{array}{ll} {}^g\mathbf{N}_1^j = {}^g_i\mathbf{R}\,{}^i\mathbf{N}_1^j & {}^i\mathbf{N}_1^j = \left[{}^i\hat{z}^j, 0, -{}^i\hat{x}^j\right]^\top \\ {}^g\mathbf{N}_2^j = {}^g_i\mathbf{R}\,{}^i\mathbf{N}_2^j & {}^i\mathbf{N}_2^j = \left[0,{}^i\,\hat{z}^j, -{}^i\hat{y}^j\right]^\top \end{array} $$

are the vectors normal to the vector from camera i's origin to the j-th target's estimated position ${}^i\hat{\mathbf{p}}^j$. Let us define matrix ${}^g\mathbf{N}^{j^\top}$ as follows:

$$ {}^g\mathbf{N}^{j^\top} = \left[\begin{array}{c} {}^g\mathbf{N}_1^{j^\top} \\ {}^g\mathbf{N}_2^{j^\top} \end{array}\right] \tag{12}$$

Thus, the observation matrix can be written as:

$$ \mathbf{H}_i^j = \frac{F_i}{({}^i\hat{z}^j)^2}{}^g\mathbf{N}^{j^\top} \tag{13}$$

IV. Designing the Value Function V

This section discusses the value function $V(\mathbf{a})$ and the properties it should possess. The objective is to allow distributed optimization over the camera network to select camera parameters $\mathbf{a}$ such that this value function is maximized. The design of the value function should first ensure that all targets are tracked at all times, while encouraging high resolution imagery at instants of time when they are possible without sacrificing the tracking specification. The value function will be the sum of two terms.

1) **Tracking**: The first term, formulated via the Fisher Information matrix, will be monotonically increasing with the tracking accuracy of the target that is least accurately tracked.
2) **Imaging**: The second term is a function of the weighted resolution of the target imagery. This second term is premultiplied by a scalar that is near zero until the tracking accuracy for all targets exceeds a user-defined threshold $\bar{P}$.

A. Fisher Information

When the target state estimation process completes at t_δ, a prior position estimate ${}^g\hat{\mathbf{p}}^j(k+1)^-$ is available for the j-th target at the future image sample time t_{k+1}, along with a prior covariance matrix $\mathbf{P}^j(k+1)^-$. In the remainder of this section, all covariance and information matrices are computed at t_{k+1}. The time argument is dropped to simplify the notation. The posterior information matrix is denoted as $\mathbf{J}^{j+} = \left(\mathbf{P}^{j+}\right)^{-1}$ which is a function of the camera settings $\mathbf{a}$:

$$ \mathbf{J}^{j+} = \mathbf{J}^{j-} + \sum_{i=1}^{N_C} \mathbf{H}_i^{j^\top}\left(\mathbf{C}_i^j\right)^{-1}\mathbf{H}_i^j \tag{14}$$

because each $\mathbf{H}_i^j$ is a function of $\mathbf{a}_i$, as was shown in Section III-D. Note also that, through $\mathbf{H}_i^j$, the posterior information is a function of the target position which is a random variable ${}^g\mathbf{p}^j \sim \mathcal{N}({}^g\hat{\mathbf{p}}^j, \mathbf{P}^{j-})$; therefore, $\mathbf{J}^{j+}$ is a random variable. Finally, note that $\mathbf{C}_i^j$ is finite only when T^j is within the field-of-view of C_i; otherwise the corresponding term of the summation has value zero.

Eqn. (14) can be decomposed as:

$$ \mathbf{J}^{j+} = \left(\mathbf{J}^{j-} + \mathbf{H}_{-i}^{j^\top}\left(\mathbf{C}_{-i}^j\right)^{-1}\mathbf{H}_{-i}^j\right) + \mathbf{H}_i^{j^\top}\left(\mathbf{C}_i^j\right)^{-1}\mathbf{H}_i^j. $$

This decomposition is convenient for decentralized optimization, because while C_i is optimizing its parameters $\mathbf{a}_i$, the contribution from prior information and all other cameras (term in parenthesis) is constant.

After optimizing $\mathbf{a}_i$, C_i broadcasts its parameter settings to its neighbors which propagate them through the network. In this manner, while any camera is locally optimizing its settings, it is accounting for both the prior information and the currently best settings of all the other cameras.

Note that in all summations in this section the information $\mathbf{H}_i^{j^\top}\left(\mathbf{C}_i^j\right)^{-1}\mathbf{H}_i^j$ for T^j from C_i is only actually received if the actual position of T^j at the time of the next image is within the field-of-view (FOV) of C_i in the next image. In the subsequent sections, the phrase "if in FOV" will be used to succinctly indicate this fact.

B. Utility $U(\boldsymbol{a})$

The parameter settings $\mathbf{a}$ determine the Fisher information and the FOV for each camera. Define a tracking utility $U_T^j(\mathbf{a})$ as:

$$ U_T^j(\mathbf{a}) = \min\left(diag\left(\mathbf{J}^{j+}\right)\right). \tag{15}$$

In addition, define:

$$\theta = \min_{j} \left(U_T^j(\mathbf{a}) \right) \tag{16}$$

$$\bar{j} = argmin \left(U_T^j(\mathbf{a}) \right) \tag{17}$$

The symbol θ is the information about least accurately tracked coordinate over all targets. The integer $\bar{j}$ is the index of that target.

The utility function is defined as:

$$U(\mathbf{a}) = \sum_{j=1}^{N_T} \left(U_T^j(\mathbf{a}) + g(\theta) w^j U_I^j(\mathbf{a}) \right). \tag{18}$$

In this definition, $U_I^j(\mathbf{a})$ is a function that rewards high resolution imagery of T^j, w^j is a possibly time varying weight that magnifies the importance of imagery for certain targets relative to others, and g is a continuously differentiable monotonically increasing bounded function such as $g(\theta) = \frac{1}{1+\exp\left(\lambda\left(\bar{P}-\theta\right)\right)}$. Such a choice of g, for large λ, ensures that the maximization of $U_I^j(\mathbf{a})$ for any target is only factored in if all coordinates of all targets are expected to exceed the accuracy specified by $\bar{P}$.

Assuming quality of image capture to be a function of the number of pixels on the target being imaged, it is desirable to have $U_I^j(\mathbf{a})$ as a monotonically increasing function but only until an imaging threshold $\bar{r}(\mathbf{a})$ is met. Let the threshold $\bar{r}(\mathbf{a})$ be a function of the maximum number of pixels permissible on the target in the image for efficient target recognition. Subsequently, $U_I^j(\mathbf{a})$ should monotonically decrease. Various choices are possible for $U_I^j(\mathbf{a})$ depending on the desired behavior. One will be considered in the implementation section.

In many instances, only one high-resolution image per target is required. Once one such image is acquire for T^j, then w^j can be set to zero so that high-resolution imagery for T^j has no added value in the future.

C. Bayesian Value $V(\boldsymbol{a})$

Because the utility $U(\mathbf{a})$ that is actually received is dependent on the random variables ${}^g\mathbf{p}^j(k+1)$ for $j = 1, \ldots, N_T$, through $\mathbf{H}_i^j$ and the FOV, the utility is a random variable. Therefore, the optimization will be based on the Bayesian value function:

$$\begin{aligned} V(\mathbf{a}) &= E\left\langle U(\mathbf{a}; {}^g\mathbf{p}^j,\ j = 1, \ldots, N_T) \right\rangle \qquad (19) \\ &= \int \left(\sum_{j} \left(U_T^j(\mathbf{a}) + g(\theta) w^j U_I^j(\mathbf{a}) \right) \right) p_{\mathbf{p}}\left(\boldsymbol{\zeta}\right) d\boldsymbol{\zeta} \end{aligned}$$

The dummy variable $\boldsymbol{\zeta}$ is used for integration over the ground plane and $p_{\mathbf{p}}$ is the Normal distribution $\mathcal{N}({}^g\hat{\mathbf{p}}^j, \mathbf{P}_{\mathbf{pp}}^{j-})$ of the predicted position of T^j in the global frame at the next imaging instant, where $\mathbf{P}_{\mathbf{pp}}^{j-}$ represents the position covariance matrix. Note that, $U_T^j(\mathbf{a})$ must account for FOV, as discussed after Eqn. (14).

V. Biased Probabilistic Camera Ordering

Each camera will optimize their own camera parameters $\mathbf{a}_i$ by maximizing the Bayesian Value Function $V(\mathbf{a})$ defined in Eqn. (19). The next camera to perform optimization will be randomly selected in a manner to favor the camera that is expected to be able to make the largest improvement to the target that is currently tracked the worst.

The posterior Information matrix for the j-th target $\mathbf{J}^{j+}$ can be represented in block form as:

$$\mathbf{J}^{j+} = \begin{bmatrix} \mathbf{J}_{\mathbf{pp}}^{j+} & \mathbf{J}_{\mathbf{pv}}^{j+} \\ \mathbf{J}_{\mathbf{vp}}^{j+} & \mathbf{J}_{\mathbf{vv}}^{j+} \end{bmatrix} \tag{20}$$

where $\mathbf{J}_{\mathbf{pp}}^{j+}$ represents the position information matrix. Using Singular Value Decomposition, the position information $\mathbf{J}_{\mathbf{pp}}^{\bar{j}+}$ of the worst tracked target can be factored as:

$$\begin{aligned} \mathbf{J}_{\mathbf{pp}}^{\bar{j}+} &= \mathbf{M}\boldsymbol{\Sigma}\mathbf{M}^\top \qquad (21) \\ &= \begin{bmatrix} \mathbf{m}_1 & \mathbf{m}_2 \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix} \begin{bmatrix} (\mathbf{m}_1)^\top \\ (\mathbf{m}_2)^\top \end{bmatrix} \qquad (22) \end{aligned}$$

In this factorization, $\mathbf{m}_1$ and $\mathbf{m}_2$ are orthonormal *information* vectors and σ_1 and σ_2 are the information in the directions of $\mathbf{m}_1$ and $\mathbf{m}_2$, respectively. Since we have assumed that all targets lie on the ground, we care only about horizontal uncertainty. From Eqns. (12) and (13), we use the horizontal component ${}^g\mathbf{N}_1^{\bar{j}}$ of $\mathbf{H}_i^{\bar{j}}$, for $i = 1, \cdots, N_C$ to define a set of scalars $\alpha_i^{\bar{j}}$ as:

$$\alpha_i^{\bar{j}} = \left| \left({}^g\mathbf{N}_1^{\bar{j}} \right)^\top \cdot \mathbf{m}_2^{\bar{j}} \right| \tag{23}$$

The scalar $\alpha_i^{\bar{j}}$ measures the alignment of C_i's horizontal observation vector with $T^{\bar{j}}$'s worst information vector.

The vector $\boldsymbol{\alpha}_{-i}^{\bar{j}} = [\alpha_1^{\bar{j}}, \ldots, \alpha_{i-1}^{\bar{j}}, \alpha_{i+1}^{\bar{j}}, \ldots, \alpha_{N_c}^{\bar{j}}]$ is normalized as:

$$\boldsymbol{\beta} = \frac{\boldsymbol{\alpha}_{-i}^{\bar{j}}}{\|\boldsymbol{\alpha}_{-i}^{\bar{j}}\|_1} \tag{24}$$

to be a probability vector (i.e. $\|\boldsymbol{\beta}\| = 1$). Given $\boldsymbol{\beta} = [\beta_1, \ldots, \beta_{N_C-1}]$, define the partition of $[0, 1]$ as:

$$\mu_l = \sum_{n=1}^{l} \beta_n$$

with $\mu_0 = 0$, and $\mu_{N_C} = 1$. A uniform random number on $[0, 1]$ will have probability β_l of being in $[\mu_{l-1}, \mu_l]$, for $l = 1, \ldots, N_C - 1$. This interval selects the index, not equal to i, for the next camera to perform optimization, in a manner that biases the selection towards those cameras with the best ability to improve $U_T^{\bar{j}}$ for the worst tracked target $T^{\bar{j}}$.

In a sequentially-ordered network, convergence to an optimum would greatly depend on the number of cameras in the network. In a 'best-camera' approach, the camera with the highest probability will always be selected to optimize next, and may lead to ignoring agents in the network that might be able to add more value. Thus, a probabilistic camera ordering mechanism is proposed. It should be noted that calculation of μ_l is independent of the potential camera settings that might be selected by the camera but is dependent on the prior Fisher information and the current settings $\mathbf{a}$ along with the camera to target normal vector, which is completely independent of $\mathbf{a}$. The camera with the largest β may or may not currently be looking at the target for the given settings $\mathbf{a}$. If selected to optimize next, it may or may not select settings that image the target.

Optimization stops when either an optimum is achieved, a user-defined stopping condition is met, or as shown in Fig 2, the time interval $t \in [t_\delta, t_\epsilon]$ allotted for optimization elapses. After optimization, cameras reconfigure themselves in the time interval $t \in [t_\epsilon, t_{k+1}]$, in readiness for upcoming images at t_{k+1}.

VI. Implementation

This section describes an implementation of the procedure proposed in this article, implemented in a simulation in MATLAB.

A. Imaging Utility $U_I^j(\mathbf{a})$

Let us define the imaging utility $U_I^j(\mathbf{a})$ as:

$$U_I^j(\mathbf{a}) = \exp\left(-\left(\frac{U_T^j(\mathbf{a}) - \bar{r}}{\sigma_r}\right)^2\right) \tag{25}$$

where $\bar{r}$ is a user-defined parameter to provide a measure of the quality of image capture and σ_r is a parameter defining the width of acceptable variation about $\bar{r}$. As future research, we will define the imaging utility as a function of the number of pixels captured on the target. With all components now defined, each camera can now apply an optimization algorithm of our choosing to maximize $V(\mathbf{a})$.

B. Optimization

Assume that it is C_i's turn to optimize first. C_i receives camera parameters $\mathbf{a}_{-i}$. It uses its existing parameters $\mathbf{a}_i$, and incoming parameters $\mathbf{a}_{-i}$, to compute Eqn. (14) and then optimizes parameters $\mathbf{a}_i$, with parameters $\mathbf{a}_{-i}$ staying constant. The sequence in which the N_C cameras optimize between $t \in [t_\delta, t\epsilon]$, is biased using the probabilistic ranking procedure described in section V. We use the Golden Section Search method [19], which is a one dimensional optimization algorithm for optimization.

C. Results

For the purpose of simulation, we assumed an area of 144 m^2 being covered by $N_C = 3$ calibrated cameras with positions:

$$C_1 = [6, 0, 5]^\top \quad C_2 = [0, 6, 5]^\top \quad C_3 = [12, 6, 5]^\top$$

Using position estimates of $N_T = 5$ targets located randomly within the area, we evaluated the system for multiple target position scenarios. In scene 1, the targets were positioned close to each other. In scene 2, they were split up in two bunches. For scene 3, the targets were placed isolated from each other, and in scene 4, one target was kept isolated from a bunch of other targets.

1) Consistency: To evaluate consistency of solutions, we started optimization $N = 100$ times, from random initial parameter settings and compare the final results. The optimum stopping conditions were $\tilde{\rho} = 1°$ and $\tilde{F} = 10^{-3}\ mm$. The results are as shown in Tables II and III, where $\bar{\rho}$ and $\bar{F}$ are the mean for the pan and focal length and σ_ρ and σ_F are the standard deviation.

TABLE II

Pan results for multiple scenarios in *degrees*

	C_1	C_2	C_3
Scenario	$\bar{\rho},\ \sigma_\rho$	$\bar{\rho},\ \sigma_\rho$	$\bar{\rho},\ \sigma_\rho$
1	8.9, 1.1	2.9, 0.8	11.3, 0.9
2	3.4, 0.8	17.5, 0.5	9.6, 1.2
3	2.7, 0.1	17.2, 1.1	9.2, 1.1
4	9.8, 0.3	6.9, 0.6	2.7, 0.7

For any scenario, all targets were always tracked to an accuracy better than $\bar{P} = 1\ m$, and when an opportunity arose, the network responded to specific highly weighted targets with high resolution imagery.

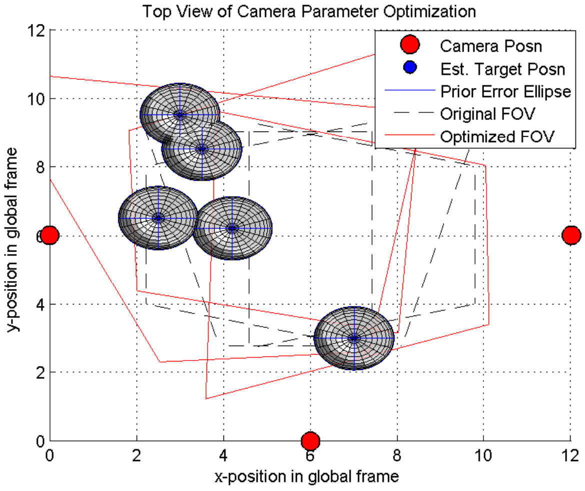

Fig. 3. Optimization for scenario 4 with $w^5 = 1$

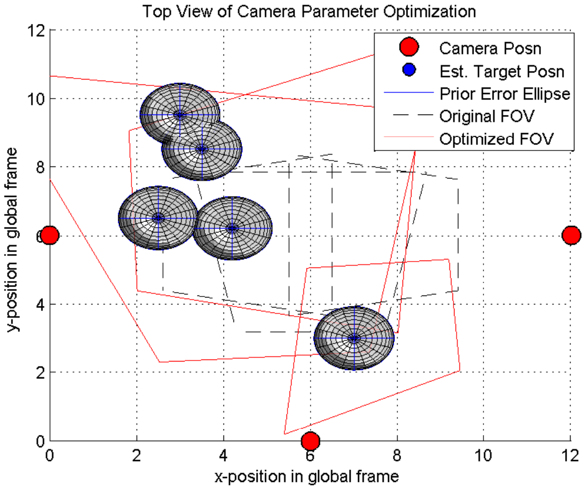

Fig. 4. Optimization for scenario 4 with $w^5 = 10$

TABLE III

FOCAL LENGTH RESULTS FOR MULTIPLE SCENARIOS IN *mm*

	C_1	C_2	C_3
Scenario	$F,\ \sigma_F$	$F,\ \sigma_F$	$F,\ \sigma_F$
1	21.4, 0.8	24.6, 0.9	23.9, 0.2
2	25.5, 0.7	22.4, 0.7	22.8, 0.6
3	22.7,0.6	23.3, 0.4	22.5, 0.1
4	22.8,0.2	22.3, 0.3	22.3, 0.1

2) Effect of weights: Assigning weights to specific targets for denoting importance of high resolution imagery has an effect on optimization. To describe this effect, let us consider scenario 4 shown in Figs. 3 and 4. These are overhead (top-view) plots of camera FOVs after optimization, for distinct weighting parameters. Targets were placed at the following locations:

$$T^1 = [2.5, 6.5, 0]^\top \quad T^2 = [4.2, 6.2, 0]^\top$$

$$T^3 = [3.5, 8.5, 0]^\top \quad T^4 = [3, 9.5, 0]^\top$$

$$T^5 = [7, 3, 0]^\top$$

For case 1, $w = [2, 2, 2, 2, 1]$, and for case 2, $w = [1, 1, 1, 1, 10]$. As can be seen in Fig. 4, due to a higher weight on T^5 in case 2, a high resolution image of T^5 was acquired by C_3, while the system maintains track on all the other targets.

VII. CONCLUSION AND FUTURE WORK

In this article, we have used a distributed network of dynamic cameras to reduce computation and communication cost, along with reduction in resources required to survey an area where the number of targets is time-varying. We propose a method for a distributed camera network to co-operatively track all targets and procure high resolution images when the opportunity arises. Distributed optimization within a Bayesian framework is presented. In addition, a probabilistic method based on singular vectors of the Fisher information matrix for biasing the random camera selection process is suggested.

As future research, due to co-operative behavior between cameras, a game-theoretic framework could be fruitful. The problem could be formulated as a game between a set of cameras competing with a set of targets, where the camera network scores points when it captures high resolution images of targets in the area, over the duration of the game. The set of targets score points every time the camera network loses track on any target. Each camera tries to co-operatively attain the maximum global utility. Formulating the problem as a potential game and using existing convergence proofs available in game-theory make this framework resourceful.

Additional interesting work could include designing a dynamic weighting scheme for targets, since the importance of a target may drastically reduce, once its high resolution image has been captured by the network.

At UC Riverside, we possess a real-life distributed camera network test-bed, where testing for the approach described in this article will be carried out in the near future. Immediate short term goals are to design an Imaging Utility as a function of image resolution, followed by application of the approach on maneuvering targets. Designing a value function to enforce continuity of optimum parameters versus time, is critical to minimize mechanical wear of the camera. Depending on the target's position, orientation and direction of motion, an aspect angle utility describing the quality of image captured by the cameras can also be included in the value function. Utilities based on target activity is another intriguing facet that can be explored, along with exploration of other quality-oriented utilities.

REFERENCES

[1] R. Olfati-Saber and N. F. Sandell, "Distributed tracking in sensor networks with limited sensing range," *Proceedings of the American Control Conference*, June 2008.

[2] R. Olfati-Saber, "Kalman-consensus filter: Optimality, stability, and performance," *2009 joint 48th IEEE Conf. on Decision and Control and 28th Chinese Control Conf.*, pp. 7036–7042, Dec 2009.

[3] B. Song and A. Roy-Chowdhury, "Stochastic Adaptive Tracking in a Camera Network," in *IEEE Intl. Conf. on Computer Vision*, 2007.

[4] C. Soto, B. Song, and A. K. Roy-Chowdhury, "Distributed multi-target tracking in a self-configuring camera network," in *Computer Vision and Pattern Recognition*, August 2009, pp. 1486–1493. [Online]. Available: http://dx.doi.org/10.1109/CVPRW.2009.5206773

[5] Y. Aloimonos, I. Weiss, and A. Bandyopadhyay, "Active Vision," in *International Journal of Computer Vision*, 1988.

[6] R. Bajcsy, "Active Perception," *Proc. IEEE*, pp. 996–1005, August 1988.

[7] D. Ballard, "Animate Vision," *Artificial Intelligence*, vol. 48, no. 1, pp. 57–86, February 1991.

[8] Q. Cai and J. Aggarwal, "Automatic Tracking of Human Motion in Indoor Scenes Across Multiple Synchronized Video Streams," in *IEEE Intl. Conf. on Computer Vision*, 1998, pp. 356–362.

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

[10] S. Khan, O. Javed, Z. Rasheed, and M. Shah, "Human Tracking in Multiple Cameras," in *IEEE Intl. Conf. on Computer Vision*, 2001, pp. I: 331–336.

[11] Y. Li and B. Bhanu, "Utility-based dynamic camera assignment and hand-off in a video network," *IEEE/ACM Intl. Conf. on Distributed Smart Cameras*, 2008.

[12] D. Markis, T. Ellis, and J. Black, "Bridging the Gap Between Cameras," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2004.

[13] K. Tieu, G. Dalley, and W. Grimson, "Inference of Non-Overlapping Camera Network Topology by Measuring Statistical Dependence," in *IEEE Intl. Conf. on Computer Vision*, 2005.

[14] H. Medeiros, J. Park, and A. Kak, "Distributed object tracking using a cluster-based kalman filter in wireless camera networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 2, no. 4, pp. 448–463, Aug. 2008.

[15] F. Qureshi and D. Terzopoulos, "Surveillance in Virtual Reality: System Design and Multi-Camera Control," *IEEE Conf. on Computer Vision and Pattern Recognition*, 2007.

[16] G. Arslan, J. Marden, and J. Shamma, "Autonomous Vehicle-Target Assignment: A Game-Theoretical Formulation," *ASME Journal of Dynamic Systems, Measurement and Control*, vol. 129, no. 5, September 2007.

[17] Farrell, J.A., "Aided Navigation: GPS with High Rate Sensors," pp. 72–75, 2008.

[18] E. Trucco and A. Verri, "Introductory Techniques for 3-D Computer Vision," pp. 26–27, 1998.

[19] Press, W.H. and Teukolsky, S.A. and Vetterling, W.T. and Flannery, B.P., "Numerical Recipes in C: second edition," pp. 397–402, 1992.