MULTI-TARGET TRACKING THROUGH OPPORTUNISTIC CAMERA CONTROL IN A RESOURCE CONSTRAINED MULTIMODAL SENSOR NETWORK

*Jayanth Nayak**

Mayachitra Inc,
5266 Hollister Ave, Ste 229,
Santa Barbara, CA 93111, USA
nayak@mayachitra.com

Luis Gonzalez-Argueta, Bi Song,
Amit Roy-Chowdhury, Ertem Tuncel

Department of Electrical Engineering,
University of California, Riverside,
CA 92521, USA
{largueta,bsong,amitrc,ertem}@ee.ucr.edu

ABSTRACT

While wide-area video surveillance is an important application, it is often not practical, from a technical and social perspective, to have video cameras that completely cover the entire region of interest. For obtaining good surveillance results in a sparse camera networks requires that they be complemented by additional sensors with different modalities, their intelligent assignment in a dynamic environment, and scene understanding using these multimodal inputs. In this paper, we propose a probabilistic scheme for opportunistically deploying cameras to the most interesting parts of a scene *dynamically* given data from a set of video and audio sensors. The audio data is continuously processed to identify interesting events, e.g., entry/exit of people, merging or splitting of groups, and so on. This is used to indicate the time instants to turn on the cameras. Thereafter, analysis of the video determines how long the cameras stay on and whether their pan/tilt/zoom parameters change. Events are tracked continuously by combining the audio and video data. Correspondences between the audio and video sensor observations are obtained through a learned homography between the image plane and ground plane. The method leads to efficient usage of the camera resources by focusing on the most important parts of the scene, saves power, bandwidth and cost, and reduces concerns of privacy. We show detailed experimental results on real data collected in multimodal networks.

Index Terms— audio-video tracking, camera control, homography estimation, graph matching

1. INTRODUCTION

Analysis of video data collected over a network of cameras covering a large geographical area has been gaining importance as a research problem. However, economic, technical and social concerns inhibit the deployment of large numbers of cameras in many application scenarios. In these situations, other kinds of sensors, e.g., audio sensors, could be deployed in addition to cameras. This would require designing efficient processing and control algorithms so that the multimodal sensors could function *collaboratively* to extract the maximum amount of information from the scene. Specifically, given the physical locations of a set of cameras and audio sensors, we would want the cameras to focus on the most interesting parts of the scene *dynamically* so that the maximum amount of data could be available for these portions.

In this paper, we focus on developing algorithms for processing in and control of such an active network of audio and video sensors and make the following specific contributions.

- Opportunistic assignment of cameras to the most interesting parts of the scene: We develop a probabilistic control algorithm based on analysis of the audio and/or video data collected over the entire network. This algorithm will be able to switch the cameras on and off, as well as change the camera parameters (pan/tilt/zoom), so as to focus the video on the most interesting parts of the scene.

- Tracking multiple targets sensed disjointly in time by audio and video sensors: We propose a multi-target tracking algorithm for objects that will usually be sensed separately by the audio and video sensors over a time interval. Note that in the proposed framework, tracking and control work together. The tracking algorithm is an inference strategy that fuses the tracks obtained individually from the audio and video inputs.

- Real-life experimental evaluation: We perform detailed experimentation on a real-life multimodal network that shows both the control and processing aspects of our proposed algorithm. Our framework for this evaluation is an intelligent building scenario with a large number of sensors and a few video cameras at critical locations.

*The first author performed the work while at UC, Riverside.

This work was supported by Aware Building: ONR-N00014-07-C-0311 and the NSF CNS 0551719. Bi Song and Amit Roy-Chowdhury were additionally supported by NSF-ECCS 0622176 and ARO-W911NF-07-1-0485.

978-1-4244-2665-2/08/$25.00 ©2008 IEEE

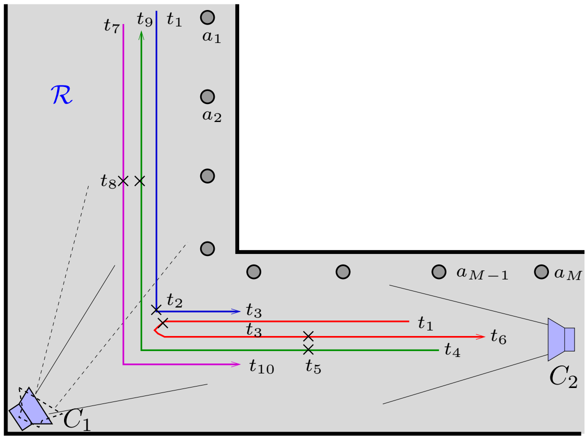

Fig. 1. An example scenario where audio can be used to efficiently control two video cameras. There are four tracks that need to be inferred. Directly indicated on tracks are time instants of interest, i.e., initiation and end of each track, mergings, splittings, and cross-overs. The mergings and cross-overs are further emphasized by X. Two innermost tracks coincide in the entire time interval (t_2, t_3). The cameras C_1 and C_2 need to be panned, zoomed, and tilted as decided based on their own output and that of the audio sensors $a_1, \ldots, a_M$.

1.1. Motivation

While video cameras for surveillance applications are now omnipresent, they consist mostly of passive sensors that are not controlled based on analysis of the collected data. This means that large numbers of such cameras need to be installed to cover the entire geographic area at all instants of time.

However, there are a number of problems in putting up too many cameras in many application scenarios. The problems stem from the cost of buying, installing and maintaining large numbers of cameras, the communication and processing burdens imposed due to transmission, storage and analysis of the video, and the social concerns associated with having too many cameras around. As a result, many applications work with only a few cameras that do not produce videos of the desired quality. A good example is a convenience store where the captured videos are usually from a specific angle and of a low resolution. Another example is surveillance videos in office buildings that are usually captured from a few pre-determined views and often have large “blind” areas.

Many of the above concerns can be addressed if we could work with only a few cameras (placed conveniently based on the application), but were able to control them efficiently so that attention could be paid to the most interesting parts of the scene. A multimodal network, where audio sensors direct the cameras to focus on the most interesting parts of the scene, is one such strategy. It involves *network-centric* processing of the audio data to determine where and when to focus the cameras. This would be followed by analysis of the video which would determine the camera parameters, i.e., whether to pan/tilt/zoom. Finally, integration of the audio and video will be needed so that we have a complete understanding of all the events in the region. Fig. 1 shows an example scenario on an L-shaped corridor where neither of the two cameras can view the entire area, and hence can infer all four tracks. However, audio data can be utilized to assign the cameras to tracks, to control their pan/tilt/zoom parameters, and most importantly, to turn them on and off intelligently. In fact, our experimental analysis is based on this very scenario, as discussed in Section 6.

1.2. Relation to Previous Work

We want to make some important distinctions between our work and some existing methods.

First, our proposed strategy is not a fusion of *simultaneous* audio and video data for better tracking, unlike [1, 2]. The audio and video inputs that we analyze contribute to tracking, for the most part, over *disjoint* time intervals. The video is switched on during *interesting* events where there is a possibility of addition or deletion of tracks (e.g., cross-overs, meetings, merging, splitting).

Second, our proposed scheme is also very different from problems like source localization and scene geometry inference using a collection of audio and video sensors [3, 4, 5, 6]. Our goal is to opportunistically assign cameras to interesting parts of the scene and free them up (to be used possibly in other tasks) when they are not needed. For example, in Fig. 1, shortly after tracks are initiated and high-resolution pictures of the subject(s) are obtained, the cameras are released until either a cross-over or merging occurs, at which point video-based tracking is needed for a short while. In the remaining times, the subject(s) can be tracked based on the audio sensors. If the cameras were to continuously follow the subjects based on the audio input, they would not have been able to view new objects that may have appeared in a different part of the scene. For example, if camera C_1 followed the track initiated at time t_4 continuously after it crossed over with another track at t_5, it would miss the initiation of another track at t_7. Camera reuse becomes even more necessary as the area being observed and the density of tracks increases.

A third area that this work touches upon is processing in networks of vision sensors [7, 8, 9, 10, 11, 12]. However, all these methods assume that it is feasible to deploy large numbers of cameras to monitor the scene and do not deal with the issue of actively controlling the cameras based on feedback from the sensed data. Our application scenario is an environment where lots of cameras cannot be deployed, and, thus, control of the few cameras available is a must. It is different from the problem of initial camera deployment [13] since the entire area is never covered by the cameras. Also, camera control is effected, unlike [14]. It is different from tracking in a dense camera network like [12].

Finally, the problem in this paper is also related to two classical problems in robotics and computer vision - sensor planning and active vision [15, 16, 17, 18]. Our work falls within the broad definition of active vision, but the novelty lies in the fact that control is achieved by analyzing audio and video data from dynamic scenes in a *network-centric* manner.

2. PRECISE PROBLEM FORMULATION

The general problem of tracking using audio and video sensors can be described as follows: the region of interest is a subset of the two dimensional plane $\mathcal{R}$. For example, $\mathcal{R}$ may represent the corridors in the floor plan of a building. Audio sensors are widely distributed across $\mathcal{R}$. Also present, but *far fewer in number*, are active video cameras that possess the ability to pan, tilt or zoom. Using these sensors, we wish to track the flow of multiple people walking through $\mathcal{R}$. More precisely, there is a set of M audio sensors $\mathcal{A} = \{a_1, \ldots, a_M\}$ distributed across $\mathcal{R}$. The audio sensors are always on and each transmits the average sound energy observed every T_a seconds to the base station (see Section 5). Note that, due to communication constraints, the sensors may only send an aggregate quantity like the energy and not the entire audio waveform. We assume that the audio sensors are synchronized. The region $\mathcal{R}$ is also observable from a set of controllable cameras $\mathcal{C} = \{C_1, \ldots, C_L\}$. Although every point in $\mathcal{R}$ is visible from at least one camera with appropriate pan/tilt/zoom (P/T/Z) parameters, there need not exist a configuration of cameras that covers every point in $\mathcal{R}$. The cameras when turned on transmit a frame every T_v seconds and are also synchronized.

The system consisting of audio sensors and cameras is used to track the motion of people in $\mathcal{R}$. While audio sensors enable efficient use of the cameras, audio data is imprecise. For example, audio data cannot reliably convey whether a single person forms a track or if there are more people walking together as a group. The tracks inferred from audio data, which we shall refer to as **a-tracks** designate paths followed by clusters composed of one or more people. Another shortcoming of audio data is that when tracks intersect, information about the identity of persons forming the a-tracks is lost. The goal of our tracking system is to track individuals, i.e., the system output is the set of paths in time and space used by people passing through $\mathcal{R}$ (see Section 4), which we term **p-tracks**. Each p-track corresponds to a unique person. The key elements of the tracking system that are dedicated to avoiding p-track ambiguity and resolving the ambiguities that do occur are:

1. *Camera Control*: After each audio or video observation, the camera control mechanism decides which cameras should be turned on or off. For cameras that are on or will be turned on, the camera control mechanism further decides what the P/T/Z parameters should be. The objective of camera control is to deploy cameras where they are needed the most, i.e., where relying on audio data would result in ambiguities such as when a-tracks split or intersect or when new a-tracks emerge. Camera control is discussed in Section 3.
2. *Person matching*: Each time we switch from audio to video tracking, the person matching algorithm finds correspondences between persons in the current view and existing p-tracks or declares the presence of someone new. The decision is based on both a-tracks and p-tracks as discussed in Section 4.

2.1. System Overview

At every instant, the multimodal sensor network control system (see Fig. 2) that we use could receive either a set of audio sensor observations and/or a set of frames of video data. When the input is audio, the observations are clustered to give estimates of the number of distinct a-tracks and their current positions in $\mathcal{R}$ (as described in Section 5.1). Following this, each cluster center is matched with an existing a-track. Each a-track is represented using a Kalman filter [19]. The state of the Kalman filter is the position and velocity of an a-track and the observation is the corresponding cluster center. More precisely, we use the following linear dynamic model for the a-track k:

$$S^k(n+1) = FS^k(n) + U^k(n) \tag{1}$$

$$O_a^k(n) = HS^k(n) + W^k(n) \; , \tag{2}$$

where the state $S^k(n) \triangleq \begin{bmatrix} X^k(n) & V^k(n) \end{bmatrix}$ with $X^k(n)$ and $V^k(n)$ denoting the position and velocity of cluster k. $F = \begin{bmatrix} 1 & T_a \\ 0 & 1 \end{bmatrix}$, $H = \begin{bmatrix} 1 & 0 \end{bmatrix}$, $U^k(n)$ and $W^k(n)$ are sequences of independent Gaussian random variables. Each Kalman filter is used to estimate the distribution of the position of the cluster at the next audio sampling instant, denoted $P^k_{n|n-1}(x)$. These estimates are used to predict a-track intersections and for subsequent camera control. Since different a-tracks span different time intervals, at any instant t, we denote the sample instant for a-track k by $n^k(t)$.

If there are unassigned clusters or a-tracks after the matching, we recognize the potential for something interesting to be happening, and the camera control mechanism requests video data from the appropriate cameras to resolve the discrepancies by either initializing new (a- and/or p-) tracks, terminating existing tracks or by discarding some cluster centers as erroneous. This is done by using a person detection algorithm on the received frame [20] and comparing the features (e.g., color histograms) of any detected humans with those of people in known p-tracks. If no matching p-tracks are found, tracks are initiated for each new person. A new a-track is assigned to the cluster formed by these persons. Person matching is also done when we switch from audio to video tracking in order to resolve ambiguities in the p-tracks. Person matching is necessary to ensure that we assign the correct a-tracks to p-tracks and also to detect any changes that might have occurred when video data was not available.

In video tracking, when a frame is received, it is fed to a particle filter based tracker which operates in the image plane (see Section 5.2). Since inference from video is more accurate, the tracking directly updates p-tracks. Each p-track that is visible in the frame has a corresponding particle filter whose output is the position of the person in the image plane. These image tracks are transformed to coordinates in $\mathcal{R}$ using a transformation from image plane to ground image plane as explained in Section 5.3.

3. CAMERA CONTROL STRATEGY

Video data is useful in understanding events like emergence or disappearance of tracks and intersection of tracks. We focus on developing control algorithms under the assumption that whenever an interesting event occurs, a camera is free to analyze it. More general settings, which would require scheduling strategies for multiple controllable cameras, are left for future work.

3.1. Setting Pan/Tilt/Zoom

The camera control algorithm described later specifies a point x that needs to be viewed. We use a heuristic algorithm to choose a camera and its P/T/Z parameters based on this point. We divide the ground plane into overlapping regions of roughly the same area. The regions $\mathcal{R}_i$, indexed by i are chosen such that for some values of pan P^l_i, tilt T^l_i and zoom Z^l_i, the region is just covered by camera C_l with (P^l_i, T^l_i, Z^l_i) as parameters. If multiple cameras can view the same region, we arbitrarily choose a camera that is not already tracking some other event.

The choice of point x while switching from audio to video is described in Section 3.2 and Section 3.3. When video data is used to confirm track initiation or deletion, since only short segments of video are sufficient, the parameters of the chosen camera remain constant. When a camera C_l is tracking for longer periods, the camera parameters might need to be continuously modified. The point x at any instant is chosen to be the centroid of the p-tracks being observed by C_l. Suppose C_l views region $\mathcal{R}_i$ and the point of interest x is also in $\mathcal{R}_i$, but is moving. As x approaches the perimeter of this region, the camera control modifies the parameters as follows. Among all the regions other than $\mathcal{R}_i$ that contain the point, it chooses the one in which the track is expected to be in for the longest time. This interval is estimated on the basis of the velocity of the track. To view the new region it might also be necessary to switch cameras. By using a finite set of P/T/Z parameters for each camera, we can compute and store the image plane to ground plane mapping for each view as explained in Section 5.3.

3.2. Track Initiation and Deletion

For every audio sampling instant t, define $\mathcal{B}_t$ as the set of clusters (see Sect. 5.1 on how clusters are formed), and $\mathcal{K}_{t-}$ as the set of confirmed a-tracks estimated based on observations before time t. We start by defining a complete bipartite graph with $\mathcal{B}_t$ and $\mathcal{K}_{t-}$ as the two vertex sets. The weight of an edge (b, k) is $\log P^k_{n^k(t)|n^k(t)-1}(b)$. Edges in this graph are pruned based on thresholding the weight. We first solve the maximum weight bipartite graph matching problem [21]. The choice of weight is motivated by the following observation: for any perfect matching [1] $f : \mathcal{K}_{t-} \to \mathcal{B}_t$ is $\log \prod_{k \in \mathcal{K}_{t-}} p(X^k_{n^k(t)} = f(k)|B^k_{1:n^k(t)-1})$. Therefore, if, as we assume, the tracks are independent, the optimal matching maximizes the likelihood of clusters given the tracks. For a matched cluster-track pair (b, k), the cluster center of c forms the input to the Kalman filter corresponding to k. Due to the pruning, there might be both unassigned cluster centers and unassigned a-tracks.

If some cluster center is unassigned, it means that there is possibly a new a-track. We turn on a camera l that is free with parameters (P^l_i, T^l_i, Z^l_i) from the allowed set such that the estimated cluster center lies in $\mathcal{R}_i$. By person detection methods [20], we can confirm the existence of an a-track and further obtain features (e.g. color histogram) for each person forming the a-track's cluster. The camera is turned off once the features are obtained. By comparing the features of the new cluster with the features of existing p-tracks, the controller can also deduce if the new cluster was formed by the splitting of an existing cluster. If no humans are detected in the view of the camera, the cluster is discarded as spurious.

If there is an unassigned a-track, we turn a camera toward the estimated track position. Given that the estimates up to the current time have been accurate, either the track has moved outside of $\mathcal{R}$ or the person or persons forming the track have stopped or the audio data is faulty. The camera can decide among these possibilities, again by using person detection. The set of a-tracks and p-tracks are updated by removing tracks that have ended.

If the expected location of any unassigned track is being observed by a camera that is already on, the video data can be used to resolve the mismatch. If there are other unassigned tracks or clusters that is neither being currently observed by any camera nor can be observed from any free camera however, we would need to define a queue of tracks and clusters that need a camera and prioritize future camera assignment. Handling such situations by coordinating between the cameras is a challenging problem for future work.

Track initiation during video tracking is done by periodically performing person detection on the observed frames.

[1] A matching is perfect if there are no unassigned clusters or a-tracks.

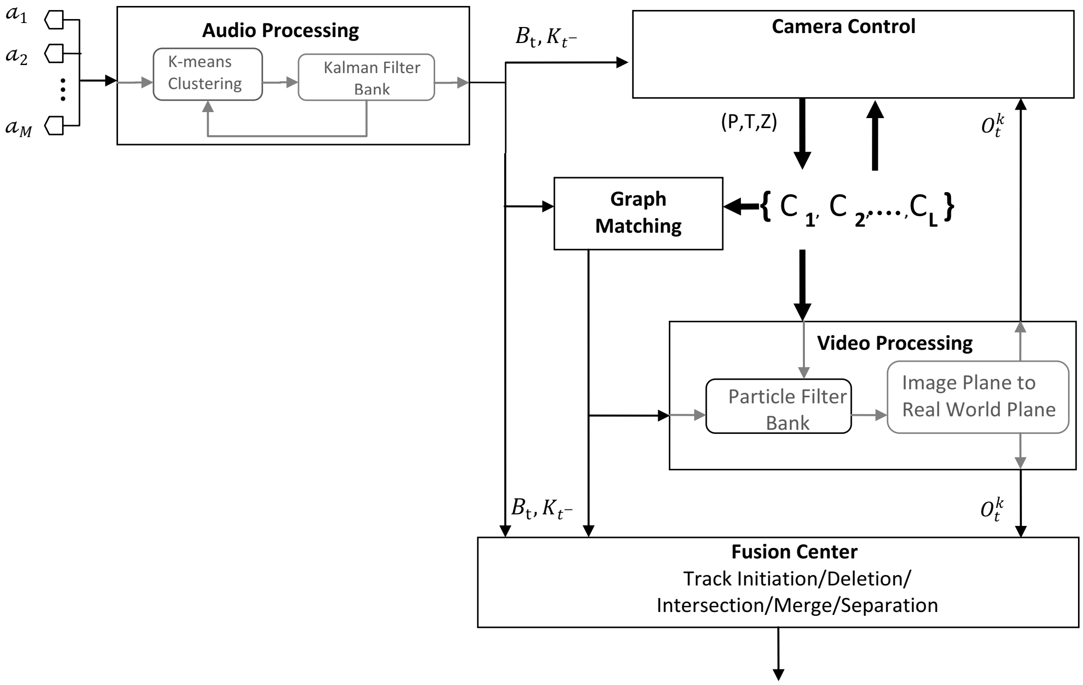

Fig. 2. Overall camera control system. Audio sensors $\mathcal{A} = \{a_1, \ldots, a_M\}$ are distributed across regions $\mathcal{R}_i$. The set of audio clusters are denoted by $\mathcal{B}_t$, and $\mathcal{K}_{t^-}$ represent the set of confirmed a-tracks estimated based on observations before time t. P/T/Z cameras are denoted by $\mathcal{C} = \{c_1, \ldots, c_L\}$. Ground plane positions are denoted by $\mathcal{O}_t^k$.

3.3. Intersecting Tracks

Intersection of p-tracks that are tracked using cameras is handled by the particle filters described in Section 5.2 which can handle occlusions. When tracking using audio, we need to consider the output of the Kalman filter for analyzing a-tracks that possibly intersect. The control decisions at observation time t are based on the quantity $P_t^{k,k'}$ defined for every pair of tracks k, k'. $P_t^{k,k'}$ is defined as the probability that the a-tracks k and k' will be within a distance threshold $D(k, k')$ at time $t + T_a$ given the observations of the a-tracks up to time t. This quantity can be estimated using the distributions $P^k_{n^k(t)+1|n^k(t)}(x)$ and $P^{k'}_{n^{k'}(t)+1|n^{k'}(t)}(x)$. The distance $D(k, k')$ depends on the extent of clusters k and k' along the shortest line in $\mathcal{R}$ joining the cluster centers. Wide clusters require a larger inter-cluster distance than narrow clusters for the clustering algorithm to resolve the audio observations into separate clusters. The controller computes $P_t^{k,k'}$ for every pair (k, k') that is not being tracked using video data. Tracking is switched to video for pairs (k, k') such that $P_t^{k,k'}$ has been increasing for $n_{a\to v}$ instants and $P_t^{k,k'} > P_{a\to v}$. To track this pair, a free camera is pointed toward the midpoint of the line joining the points where the two tracks are expected to be the closest. These points are estimated using the states of the Kalman filters.

We assume that the above algorithm has detected all tracks that will intersect and has turned the camera on before the actual intersection of a-tracks has occurred. Once the camera is turned on, the first task is to decide the mapping between a-tracks and p-tracks as described in Section 4. The camera control system can track the intersection by updating the P/T/Z parameters of the cameras that are turned on. Since we need to minimize usage of the cameras, we need a mechanism to revert to audio tracking at a suitable time. To decide whether to switch from video to audio data, we cluster the positions of the viewed p-tracks. A switch occurs in the following situations

1. *Separating clusters*: A camera is released if for n_{sep} video sample instants a) the p-tracks in its view induce the same clustering, b) the intercluster distance is above a threshold D_{sep}, and c) the intercluster distance is increasing. An a-track is initialized for each cluster of p-tracks.

2. *Merging clusters*: Tracks are merged if a) distance between every pair of p-tracks corresponding to a camera is less than D_{merge} for n_{merge} instants, and b) if the velocity of these p-tracks in the ground plane is approximately the same. The camera is then released and a single a-track is initialized for all the p-tracks.

Algorithm 1 Audio Tracking Algorithm

```
procedure AUDIOTRACK
    Read audio inputs periodically and cluster                              ▷ Section 5.1
    Match clusters with existing a-tracks and extend matched a-tracks       ▷ Section 3.2
    for all Unassigned clusters do
        VIDEOTRACK( Cluster center, INITIATE)
    end for
    for all Unassigned a-tracks do
        VIDEOTRACK( Expected a-track position, DELETE)
    end for
    for all Pairs of a-tracks expected to intersect do
        VIDEOTRACK( Expected point of intersection, INTERSECT)
    end for
end procedure
```

4. COMPUTING FINAL TRACKS OF ALL TARGETS

Whenever a camera is turned on, we need to map persons detected in the first frame to existing p-tracks or initiate new tracks. This is again done by graph matching. For each p-track we store the most recent position of that p-track for which we have image feature data. We create a bipartite graph [22] where there is an edge between a p-track and a person detected in the current frame if the distance between the corresponding features is less than a threshold and if there exists an a-track connecting the last position of the p-track and the position of the person deduced from the current frame using the image plane to ground plane mapping. The weight of an edge is the distance between the features. If after the (minimum weight) graph matching, a person in the current frame has no correspondence with any p-track, we initiate a new p-track. Also, for previous p-tracks not associated to people in the current frame, we assume that these persons have stopped somewhere along the a-track. Some ambiguity remains as to the exact position where the p-tracks stopped.

In this paper, we restrict ourselves to matching persons detected in the current frame with p-tracks assuming that the most recent observation of the p-track was correct. A more accurate procedure would attempt to match features and a-tracks along entire p-tracks and correct wrong assignments from the past. This is a task left for future work.

The system output combines the results of audio and video tracking using person matching. When a person is tracked using video, the p-tracks are directly updated. The segments where the person is tracked by using audio, i.e., the a-tracks are stitched with the p-tracks on the basis of person matching.

5. AUDIO AND VIDEO PROCESSING

5.1. Clustering of Audio Observations

The goal of the audio processing is to generate rough estimates of the current position of each trajectory. At every instant, we first compute the set S, of audio samples whose amplitude is above a threshold A_{audio}. A cluster C is formed by grouping together the positions of samples from S such that $x \in C \Leftrightarrow (d(x, C \setminus x) < D_{\text{intra}})$ and $d(x, S \setminus C) \geq D_{\text{inter}}$ ($d(\cdot, \cdot)$ is the squared error distortion measure extended to a point-set distance in the usual way). This process is repeated till all elements in S fall in some cluster. Clusters with a single element are deleted. The center of every cluster is the location of the sensor in the cluster with the highest amplitude.

5.2. Video Tracking in the Image Plane

Tracking in video is done using particle filters, the details are given below.

5.2.1. Dynamical Model

Similar to [23], we represent the target regions by ellipses. The state vector is given as $\Phi_n = [X_n^I, \dot{X}_n^I, H_n^I, \dot{H}_n^I]$, where X_n^I, $\dot{X}_n^I$ denote the center and velocity of the ellipse in the image plane, H_n^I and $\dot{H}_n^I$ denote the lengths of the half axes and the corresponding scale changes. Then the dynamics are modeled by a stationary Gauss-Markov process, i.e., ,

$$\Phi_n = A\Phi_{n-1} + Z_n, \; Z_n \sim \mathcal{N}(0, \Sigma_Z). \tag{3}$$

5.2.2. Observation Model for Tracking

The observation process is defined by the posterior distribution, $p(I_n|\Phi_n)$, where I_n is the image observation. Our observation models were generated based on the color appearance. To build a color appearance model, we sample the normalized color information [24] (reduces illumination problems) from the input image I_n over the region covered by the predicted ellipse $E_n(X_n^I, H_n^I)$ to get the color histogram ch. The observation likelihood is defined as

$$p(I_n|\Phi_n) = \gamma_a \exp\{-||\mathcal{B}(ch, CH)||^2\}, \tag{4}$$

where $\mathcal{B}(.)$ is the Bhattacharya distance between two color histograms, CH is the color histogram associated with the

Algorithm 2 Algorithm for Video Tracking and Track Fusion

```
procedure VIDEOTRACK(Position, Type)
    Start camera that can view Position                                          ▷ Section 3.1
    Get frame and perform person detection
    Match detected persons with existing p-tracks                                ▷ Section 4
    if Type=INITIATE or DELETE then                                              ▷ Section 3.2
        Use unmatched persons and p-tracks to initiate or delete p-tracks and a-tracks
    else if Type=INTERSECT then                                                  ▷ Section 3.3
        Fuse a-tracks with p-tracks                                              ▷ Section 4
        VideoTrack ← $1$
        while VideoTrack=1 do
            Get frame
            Update p-tracks using particle filters                               ▷ Section 5.2
            Cluster p-tracks
            if Clusters constant, sufficiently distant and with separating velocities for past $n_{\text{sep}}$ frames then
                Initiate a-track for each cluster
                VideoTrack ← $0$
            end if
            if All pairwise inter p-track distances small and all p-track velocities similar for past $n_{\text{merge}}$ frames then
                Initiate single a-track for all p-tracks
                VideoTrack ← $0$
            end if
            Update camera parameters based on centroid of p-tracks
            Periodically perform person detection to confirm and modify existing p-tracks
        end while
    end if
    Stop camera
end procedure
```

ellipse E_0 (the initialization generated from object of interest in the first frame) and γ_a is normalizing factor.

5.2.3. *Tracking using Auxiliary Particle Filters*

In this paper, we use an auxiliary particle filter (APF) for tracking. The APF with sampling/importance resampling was introduced by Pitt and Shephard [25] as a variant of the standard particle filter to improve sample efficiency. In a standard particle filter a collection of particles with weights is used as a discrete approximation to the posterior $p(\Phi_n|I_{1:n})$. In an APF, additionally, the branch index B, which defines a correspondence between particles at time n and those at time $n-1$, is introduced as an auxiliary variable [26]. The target joint density function is $p(\Phi_n, B|I_{1:n})$. The sampling can thus be performed in two steps. For each particle j, draw $B^{(j)}$ from the distribution $p(B|I_{1:n})$. Then draw Φ_n^j from $p(\Phi_n|\Phi_{n-1}^{B^{(j)}})$, the prior density based on $B^{(j)}$. The joint sample set is $\{\Phi_n^j, B^{(j)}\}$ and the corresponding weights are updated by $w_n^j \propto w_{n-1}^{B^{(j)}} \frac{p(I_n|\Phi_n^j)p(\Phi_n^j|\Phi_{n-1}^{B^{(j)}})}{p(\Phi_n^j, B^{(j)}|I_n)}$. These are then used to form an estimate of the posterior distribution, $p(\Phi_n, B|I_{1:n})$. We use an APF because the observation model is nonlinear and the posterior can temporarily become multimodal due to background clutter and occlusion.

5.3. Mapping from image plane to world plane

We adopted the method in [27] to map the image plane to the world plane. There exists a matrix H, called the projective transformation matrix, such that if $\vec{x}$ and $\vec{x}'$ are the homogeneous coordinate vectors denoting, respectively, a point in the world plane and its corresponding point in the image plane, then $\vec{x} = H\vec{x}'$. The matrix H has a decomposition of the form $H = SAP$, where

$$P = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ l_1 & l_2 & l_3 \end{bmatrix} \tag{5}$$

is a pure projective transformation and $\vec{l}_\infty = \begin{bmatrix} l_1 & l_2 & l_3 \end{bmatrix}^T$ specifies the vanishing line of the plane.

$$A = \begin{bmatrix} \frac{1}{\beta} & -\frac{\alpha}{\beta} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{6}$$

is an affine transformation. The parameters α and β specify the image of the circular points [28]. The final matrix in the decomposition is a similarity transformation

$$S = \begin{bmatrix} sR & \vec{t} \\ \vec{0}^T & 1 \end{bmatrix} \tag{7}$$

where R is a rotation matrix, $\vec{t}$ a translation vector, and s an isotropic scaling.

The parameters defining H can be estimated by automatic detection of vanishing points and orthogonal directions [27].

6. EXPERIMENTAL RESULTS

The camera control system was tested in the scenario depicted in Fig. 1. The region $\mathcal{R}$ was an L-shape corridor of length 40 m. Twenty microphone sensors (MICAZ wireless motes with MTS310 sensor boards) were placed along the corridor at intervals of 2 meters. Two P/T/Z wireless cameras were also placed as shown in Fig. 1. The cameras were mounted at an elevation for better coverage. A scenario with four subjects walking in $\mathcal{R}$ was used to demonstrate camera control performance during initialization, split, merge and deletion of tracks. The region $\mathcal{R}$ can be approximately equated to a line, by which both position and velocity become one dimensional quantities, thus allowing easy graphical representation, but not simplifying the essence of our approach. The experiment lasted 80s during which the tracking switched between audio and video as shown in Fig. 3 (a).

At time $t = 0$, by processing audio data, we observe that there are two new clusters, 20m apart. Camera C_1 confirms the presence of person P1 and also obtains features for this person. Camera C_2 does the same for person P2. The cameras are turned off once the features are captured. In Fig. 3, there is a unique color associated with each person and their p-tracks. At all transitions from audio to video, the colored squares show the results of feature matching, where two squares are the same color if they were chosen by the bipartite graph matching. Comparing Fig. 1 and Fig. 3 (a), we can see that the matching correctly associates features with p-tracks. Since the identity of persons is lost during tracking by audio, a-tracks are shown in black.

As the two a-tracks get closer, the probability that they will intersect crosses a threshold at around 7s (shown in Fig. 3 (b)) triggering the video tracking with C_1 covering the required area. However, for the interval between 7 and 13 s, neither person is in the chosen view of C_1 and there are no distinct audio clusters either. So, the a-tracks and p-tracks cannot be updated in this interval. Once the persons appear, graph matching is used to associate the persons detected in the frame with P1 and P2. Since P1 and P2 begin to walk together, at around time $t = 17$s, the persons near the boundary of the current view, which causes a change in the camera parameters. Bipartite graph matching is again performed after the change in camera parameters. At time $t = 22$s, based on the proximity of P1 and P2 and their relative velocities, the system concludes that the two persons will form a cluster and turns off C_1 and initializes an a-track for the cluster formed by P1 and P2.

This a-track is continued using audio data till $t = 29$ when two audio clusters are detected. C_2 detects a person and the person matching algorithm declares that it is neither P1 nor P2, but a new person P3. Audio tracking of the P1-P2 a-track and the new a-track with P3 continues till they are sufficiently close. Processing switches to video obtained from C_1 with an appropriate P/T/Z setting. Two persons are detected and they are matched to P2 and P3. We can therefore infer that P1 stopped (or went outside $\mathcal{R}$) sometime between $t = 22$ and $t = 30$.

The intersection of tracks of P2 and P3 is observed using C_2 with a new P/T/Z setting and tracked using particle filters. Once the video processing module declares that the distance between P2 and P3 has consistently remained above 3m for 5s and that their velocities are in opposite directions, the camera is turned off and processing is passed to the audio module. The two clusters observed by audio processing are associated with a-tracks initiated by P2 and P3. Soon after, P2 is observed leaving $\mathcal{R}$ and its p-track is stopped.

At $t = 49$s, a new audio cluster is observed and now C_1 is used to infer the presence of a new person P4. Once the corresponding features are stored, C_1 is turned off and the two a-tracks are updated using audio data as it arrives. Since the two persons are walking towards each other, the audio processing indicates at time $t = 55$ that the two a-tracks will soon intersect. The camera C_1 is turned on with new P/T/Z parameters to focus on the expected point of intersection. C_1 is used to track P3 and P4 until P3 crosses the boundary of $\mathcal{R}$ at around $t = 60$. The camera is turned off and the audio is used to track P4 till the end of the experiment. During this interval, there is a single a-track which is also the p-track for P4. The complete set of inferred p-tracks is shown in Fig. 4.

7. CONCLUSION

In this paper, we propose a probabilistic scheme for opportunistically deploying cameras to the most interesting parts of a scene sensed by a multimodal network. This allows us to analyze the events in the scene using a much smaller number of cameras leading to significant savings of power, bandwidth, storage and money and reduction of privacy concerns. The audio data is continuously processed to identify interesting events, e.g., entry/exit of people, merging or splitting of groups, and so on. This is used to indicate the time instants to turn on the cameras. Thereafter, analysis of the video determines how long the cameras stay on and whether their pan/tilt/zoom parameters change. Events are tracked continuously by combining the audio and video data. Correspondences between the audio and video sensor observations are obtained through a learned homography between the image plane and ground plane. We show detailed experimental results on real data collected in multimodal networks.

8. REFERENCES

[1] M. Beal, N. Jojic, and H. Attias, “A graphical model for audiovisual object tracking,” in *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2003, vol. 25, pp. 828–836.

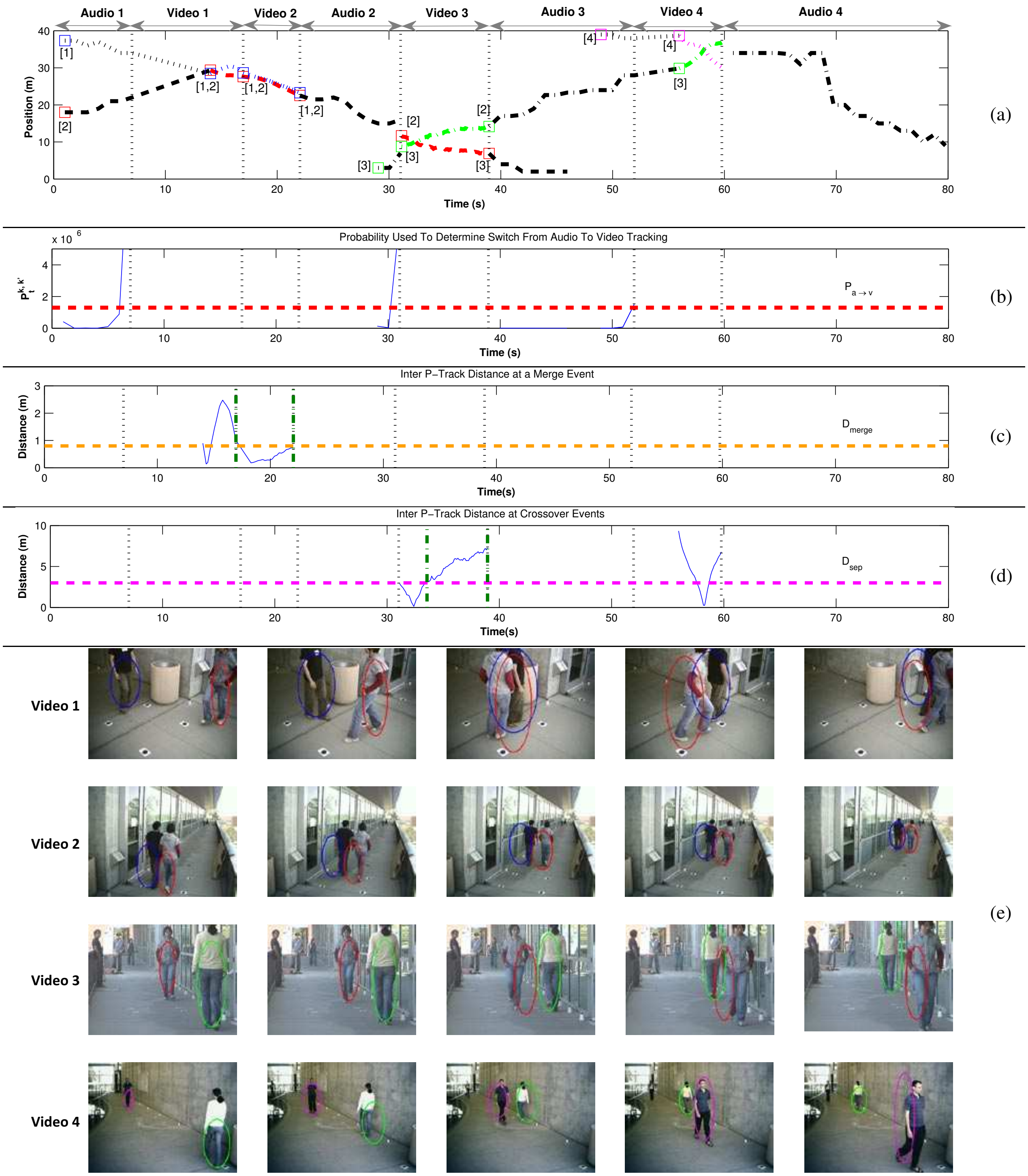

Fig. 3. Performance of the camera P/T/Z control algorithm in merge, split, initialization and deletion scenarios. The audio in (a) is displayed using black line styles since audio data does not contain information about the identity of people causing high audio amplitudes. The colored blocks in (a) represents features matched to previous information about persons: same colored squares denote that the same person is represented by the features. Also, a number is associated with each person/square for clarity. (b) shows the probability that is used to switch from audio to video in track intersection scenarios. (c) and (d) show the measures used to decide splits and merges. (e) shows some representative frames from the intervals where there is video data. The ellipses used by the particle filters are overlaid on the corresponding person.

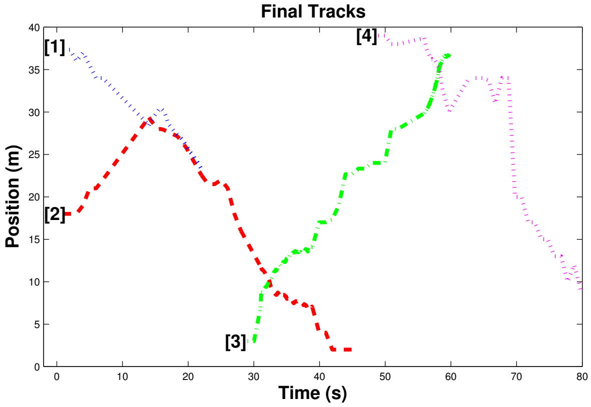

Fig. 4. Fusion results showing the full trajectory of every target in time.

[2] X. Zou and B. Bhanu, "Tracking humans using multimodal fusion," in *Proc. IEEE Workshop on Object Tracking and Classification Beyond the Visible Spectrum*, 2005.

[3] A. O'Donovan, R. Duraiswami, and J. Neumann, "Microphone arrays as generalized cameras for integrated audio visual processing," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, 2007.

[4] S. Negahdaripour, "Epipolar geometry of opti-acoustic stereo imaging," in *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2007.

[5] E. Kidron, Y. Schechner, and M. Elad, "Pixels that sound," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, 2005.

[6] Y. Ke, D. Hoiem, and R. Sukthankar, "Computer vision for music identification," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, 2005.

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

[8] D. Makris, T. Ellis, and J. Black, "Bridging the gap between cameras," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, 2004.

[9] K. Tieu, G. Dalley, and W. E. L Grimson, "Inference of non-overlapping camera network topology by measuring statistical dependence," in *International Conf. on Computer Vision*, 2005.

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

[11] S. Khan, O. Javed, Z. Rasheed, and M. Shah, "Camera Handoff: Tracking in Multiple Uncalibrated Stationary Cameras," in *IEEE Workshop on Human Motion*, 2000.

[12] B. Song and A. Roy-Chowdhury, "Stochastic adaptive tracking in a camera network," in *Proc. of International Conf. on Computer Vision*, 2007.

[13] U. M. Erdem and S. Sclaroff, "Automated camera layout to satisfy task-specific and floor plan-specific coverage requirements," *Computer Vision and Image Understanding*, vol. 103, pp. 156–169, 2006.

[14] F. Z. Qureshi and D. Terzopoulos, "Surveillance in virtual reality: System design and multi-camera control," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, June 2007, pp. 1–8.

[15] K. Tarabanis, P. Allen, and R. Tsai, "A Survey of Sensor Planning in Computer Vision," *IEEE Trans. on Robotics and Automation*, vol. 11, pp. 86–104, February 1995.

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

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

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

[19] A. H. Sayed, *Fundamentals of adaptive filtering*, John Wiley & Sons, Inc., Hoboken, NJ, 2003.

[20] B. Wu and R. Nevatia, "Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors," in *Proc. of International Conf. on Computer Vision*, 2005.

[21] J. Hopcroft and R. Karp, "An $n^{2.5}$ algorithm for maximum mathings in bipartite graphs," *SIAM J. Computing*, 1973.

[22] O. Javed, Z. Rasheed, K. Shafique, and M. Shah, "Tracking in multiple cameras with disjoint views," in *Proc. of International Conf. on Computer Vision*, 2003.

[23] K. Nummiaro, E. Koller-Meier, and L. Van Gool, "Object tracking with an adaptive color-based particle filter," in *Symp. for Pattern Recognition of the DAGM*, 2002.

[24] G. D. Finlayson, B. Schiele, and J. L. Crowley, "Comprehensive colour image normalization," in *Proc. of European Conference on Computer Vision*, 1998.

[25] M. K. Pitt and N. Shephard, "Filtering via simulation: Auxiliary particle filters," *Journal of the American Statistical Association*, vol. 94, pp. 590–599, 1999.

[26] F. Guo and G. Qian, "Sample-Efficiency-Optimized Auxiliary Particle Filter," in *Proceedings of IEEE Workshop on Statistical Signal Processing*, 2005.

[27] D. Liebowitz and A. Zisserman, "Metric rectification for perspective images of planes," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, 1998.

[28] J. Semple and G. Kneebone, *Algebraic Projective Geometry*, Oxford University Press, 1979.