Multiple Camera Fusion for Multi-Object Tracking

Shiloh L. Dockstader[†] and A. Murat Tekalp
Department of Electrical and Computer Engineering
University of Rochester, Rochester, NY 14627 USA
URL: http://www.ece.rochester.edu/~dockstad/research/

Abstract

We propose a distributed, real-time computing platform for tracking multiple interacting persons in motion. To overcome occlusion and articulated motion we use a multi-view implementation, where 2-D semantic features are independently tracked in each view and then collectively integrated using a Bayesian belief network with a topology that varies as a function of scene content and feature confidence. The network fuses observations from multiple cameras by resolving independency relationships and confidence levels within the graph, thereby producing the most likely vector of 3-D state estimates given the available data. We demonstrate the efficacy of the proposed system using a multi-view sequence of several people in motion. Our experiments suggest that, when compared with data fusion based on averaging, the proposed technique yields a noticeable improvement in tracking accuracy.

1. Introduction

1.1. Motivation

Motion analysis and tracking from monocular or stereo video has long been proposed for applications such as visual security and surveillance [1][2]. More recently, it has also been employed for performing gesture and event understanding [3] and developing ubiquitous and wearable computing for the man-machine interface [4]. However, due to a combination of several factors, reliable motion tracking still remains a challenging domain of research. The underlying difficulties behind human motion analysis are founded mostly upon the complex, articulated, and self-occluding nature of the human body [5]. The interaction between this inherent motion complexity and an equally complex environment greatly compounds the situation. Environmental conditions such as dynamic ambient lighting, object occlusions, insufficient or incomplete scene knowledge, and other moving objects and people are just some of the naturally interfering factors. These issues make tasks as fundamental as tracking and semantic correspondence recognition exceedingly difficult without first outlining numerous, and sometimes prohibitive, assumptions. Computational complexity also plays an important role, as many of the popular applications of human motion analysis and monitoring demand real-time (or near real-time) solutions.

1.2. Previous work

Various approaches to object tracking employ features such as edges, shape, color, and optical flow [6]. To handle occlusion, articulated motion, and noisy observations, numerous researchers have looked to stochastic modeling such as Kalman filtering [7][8] or conditional density propagation (*Condensation*) [9]. Dockstader and Tekalp [2] suggest a modified Kalman filtering approach to the tracking of several moving people in video surveillance sequences. They take advantage of the fact that as multiple moving people interact, the state predictions and observations for the corresponding Kalman filters no longer remain independent. Jang and Choi [10] suggest the use of active models based on regional and structural characteristics such as color, shape, texture, and the like to track non-rigid moving objects. The active models employ Kalman filtering to predict basic motion information and snake-like energy minimization terms to perform dynamic adaptations using the moving object's structure. MacCormick and Blake [11] present the notion of *partitioned sampling* to perform robust tracking of multiple moving objects. The underlying probabilistic exclusion principle prevents a single observation from supporting the presence of multiple targets by employing a specialized observational model. Without the explicit use of extensive temporal or stochastic modeling, McKenna *et al.* [12] describe a computer vision system for tracking

[†] Corresponding Author

0-7695-1171-6/01 $10.00 © 2001 IEEE

multiple moving persons in relatively unconstrained environments. Haritaoglu *et al.* [13] present a real-time system, W^4, for detecting and tracking multiple people when they appear in a group.

The use of multi-view monitoring [14] and data fusion [15] provides an eloquent mechanism for handling occlusion, articulated motion, and multiple moving objects in video sequences. Stillman *et al.* [16] propose a robust system for tracking and recognizing multiple people with two cameras capable of panning, tilting, and zooming (PTZ) as well as two static cameras for general motion monitoring. Utsumi *et al.* [17] suggest a system for detecting and tracking multiple persons using multiple cameras. The system is composed of multiple tasks including position detection, rotation angle detection, and body-side detection. For each of the tasks, the camera that provides the most relevant information is automatically chosen using the distance transformations of multiple segmented object maps. Cai and Aggarwal [18] develop an approach for tracking human motion using a distributed-camera model. The system starts with tracking from a single camera view and switches when the active camera no longer has a sufficient view of the object.

For a more detailed discussion of modeling, tracking, and recognition for human motion analysis, we refer the reader to the special issue of the *IEEE Trans. on Pattern Analysis and Machine Intelligence* on surveillance [19].

1.3. Proposed contribution

In this paper, we introduce a distributed, real-time computing platform for improving feature-based tracking in the presence of articulation and occlusion for the goal of recognition. The main contribution of this work is to perform both spatial and temporal data integration within a unified framework of 3-D position tracking to provide increased robustness to temporary feature point occlusion. In particular, the proposed system employs a probabilistic weighting scheme for spatial data integration as a simple Bayesian belief network (BBN) with a dynamic, multidimensional topology. For the proposed system, this corresponds to the selective use of multiple views of particular features based on measures of spatio-temporal tracking confidence.

2. Theory

2.1. System overview

The proposed system, as shown in Figure 1, consists of three major components: (i) state-based 2-D predictor-corrector filtering for monocular tracking (in the dotted box), (ii) multi-view (spatial) data fusion, and (iii) Kalman filtering for 3-D trajectory tracking. The first stage consists of video preprocessing including background subtraction, sparse 2-D motion estimation, and foreground region clustering. From the estimated 2-D motion, we extract a set of measurements (observations) $\mathbf{x}[k]$, where $k \geq 0$ is the frame number, for the estimation of the state vector, $\mathbf{s}[k] \equiv [\mathbf{s}_1[k] \;\; \mathbf{s}_2[k] \;\; \cdots \;\; \mathbf{s}_N[k]]^T$. Here, $\mathbf{s}_m[k]$, $1 \leq m \leq N$, denotes the image coordinates of the m^{th} feature point that we wish to track in time, and N is the number of features being tracked on one or more independently moving regions. We also define, $\boldsymbol{\sigma}[k]$, as a 3-D state vector that captures both the velocity and position of features in 3-D Cartesian space, where the unknown 3-D feature position is denoted by $\mathbf{y}[k]$. At each frame, k, we use the observations, $\mathbf{x}[k]$, in conjunction with 3-D state estimates, $\hat{\boldsymbol{\sigma}}[k-1 \mid k-1]$, as input to a predictor-corrector filter. The output of the predictor-corrector filter is a state estimate, $\hat{\mathbf{s}}[k]$, with some confidence, $\mathbf{M}[k]$.

The stage of multi-view fusion (indicated as the Bayesian network in Figure 1) performs spatial data integration using triangulation, perspective projections, and Bayesian inference. The input to the network is a set of random variables, $\{\xi_1[k], \xi_2[k], \cdots, \xi_J[k]\}$, where the j^{th} element is identical to the output, $\hat{\mathbf{s}}[k]|_j$, taken from the predictor-corrector filter corresponding to the j^{th} view of the scene. The network defines an unknown subset of $\{\xi_1[k], \xi_2[k], \cdots, \xi_J[k]\}$, denoted by $\xi[k]$, that combine to form yet another random variable, $\boldsymbol{\lambda}[k]$, indicative of a vector of 3-D positions. At the output of the BBN are the estimates, $\hat{\xi}[k]$ and $\hat{\mathbf{y}}[k]$, that maximize the joint density, $P_{\Lambda,\Xi}[\boldsymbol{\lambda}, \xi]$. Accompanying the 3-D estimate is a noise covariance matrix, $\mathbf{R}[k]$.

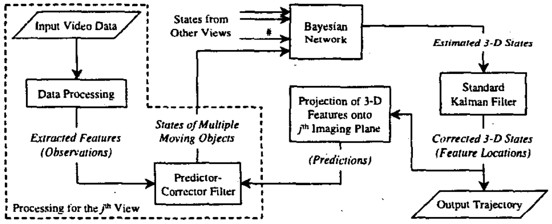

Input Video Data
Data Processing
Extracted Features (Observations)
Predictor-Corrector Filter
Processing for the j^{th} View
States of Multiple Moving Objects
States from Other Views
#
Bayesian Network
Estimated 3-D States
Standard Kalman Filter
Corrected 3-D States (Feature Locations)
Projection of 3-D Features onto j^{th} Imaging Plane
(Predictions)
Output Trajectory

Figure 1. System flow diagram.

The final stage of the system uses a Kalman filter to maintain a level of temporal smoothness on the vector of 3-D trajectories. The observations for the Kalman filter are the 3-D estimates, $\hat{\mathbf{y}}[k]$, produced by the Bayesian network. As mentioned previously, the states of the filter are indicated by $\boldsymbol{\sigma}[k]$ which represent both the true velocity and position of the N features in 3-D Cartesian space. The corrected states, $\hat{\boldsymbol{\sigma}}[k \mid k]$, at the output of the filter provide updated estimates of the unknown 3-D position vector.

Two-dimensional feature tracking is performed on a dedicated, independent processor for each camera view ($1 \leq j \leq J$). The features for each view are then passed to a central processor that integrates and filters the data using the Bayesian network and 3-D Kalman filter, respectively.

2.2. Two-dimensional feature tracking

Throughout this section it is tacitly understood that each equation is applied to data at the plane of the j^{th} camera, although we drop the explicit dependence on j with the understanding that the steps in question are equally applicable to multiple views of the scene. As in [2], the proposed technique performs moving object detection and segmentation using change detection and localized, sparse motion estimation over a grid of points, including the semantic features, within the foreground of the video sequence.

The first step involves computation of the state prediction, $\hat{\mathbf{s}}[k \mid k-1]$, in the current frame according to

$$\hat{\mathbf{s}}[k \mid k-1] = \mathsf{F}\left(\boldsymbol{\Psi}[k]\hat{\boldsymbol{\sigma}}[k-1 \mid k-1]\right), \qquad (1)$$

where $\boldsymbol{\Psi}[k]$ is a 3-D state transition matrix and $\mathsf{F}(\cdot)$ is a vector-valued function that maps points in 3-D Cartesian space to a particular imaging plane using a perspective projection. The precise definition of $\boldsymbol{\Psi}[k]$ is given in §2.4. We then compute the 2-D error covariance matrix, $\mathbf{M}[k \mid k-1]$, by projecting a 3-D error covariance matrix, $\boldsymbol{\Gamma}[k \mid k-1]$, associated with $\hat{\boldsymbol{\sigma}}[k-1 \mid k-1]$ to each imaging plane as per

$$\mathbf{M}[k \mid k-1] \triangleq \mathsf{G}\left(\boldsymbol{\Gamma}[k \mid k-1]\right), \qquad (2)$$

where $\mathsf{G}(\cdot)$ is a transformation of the noise covariance from 3-D to 2-D. The mapping of 3-D data to the imaging plane of each camera, as in (1) and (2), results in a one-step predictor-corrector filter at each frame, k. Collectively, the implementation of these functions, $\mathsf{F}(\cdot)$ and $\mathsf{G}(\cdot)$, parallels that of a transformation of random variables. An analytical method for estimating these transformations is described in §2.3.

The next step involves the computation of the gain matrix, $\mathbf{K}[k]$, which determines the magnitude of the correction at the k^{th} frame. The operation relies on $\mathbf{M}[k \mid k-1]$ and a noise covariance matrix, $\mathbf{C}[k]$, which describes the distribution of the assumed Gaussian observation noise. Using a probabilistic weighting scheme, similar to that proposed in [2], we introduce a matrix that captures the confidence associated with the temporal correspondence measurements for each feature. For this task, we classify the correspondence of various features and their neighboring pixels into three basic classes:

- *Class A* – The element is visible in the previous frame and presumably in the current frame, as there exists a *strong* temporal correspondence;
- *Class B* – The element is visible in the previous frame but presumably not in the current frame, as there exists only a *weak* temporal correspondence; and
- *Class C* – The element is not visible in the previous frame.

The qualification of a feature as having a *strong* or *weak* correspondence is based on well-established criteria in the motion estimation and optical flow literature [20].

Let us refer to the i^{th} motion vector in the neighborhood of some feature point, $\mathbf{s}_m[k]$, as $\mathbf{v}_i[k]$ and the origin of this vector in the previous frame (in image coordinates) as $\tilde{\mathbf{v}}_i[k]$. For convenience, we introduce the notation $\boldsymbol{\Omega}_\mathbf{A}$, $\boldsymbol{\Omega}_\mathbf{B}$, and $\boldsymbol{\Omega}_\mathbf{C}$ to represent the sets of all points, $\tilde{\mathbf{v}}_i[k]$ and $\hat{\mathbf{s}}_m[k-1]$, that are classified as *Class A*, *Class B*, and *Class C*, respectively. It is assumed that the union of these three sets describes a single, arbitrary region. For the proposed contribution, this corresponds to an entire moving person, although the technique is equally applicable to single body parts, groups of moving people, or even generic moving regions, depending on the segmentation of the foreground.

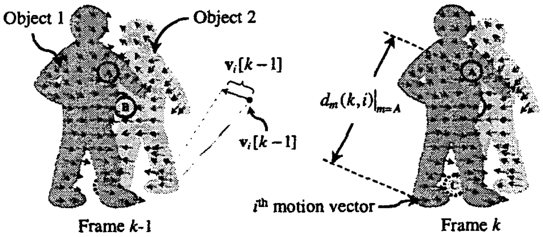


Figure 2. Occlusion modeling.

We construct $\mathbf{C}[k]$ in the traditional manner by allowing the matrix to be representative of the time-varying differences between observations and corrected predictions. We then compose a gain matrix according to

$$\mathbf{K}[k] = \mathbf{M}[k \mid k-1]\mathbf{H}^T[k]\mathbf{Z}^{-1}[k], \qquad (3)$$

where $\mathbf{H}[k] = \mathbf{I}$ represents the linear observation matrix,

$$\mathbf{Z}[k] = \mathbf{W}[k]\mathbf{C}[k]\mathbf{W}^T[k] + \mathbf{H}[k]\mathbf{M}[k \mid k-1]\mathbf{H}^T[k], \quad (4)$$

$$\mathbf{W}[k] \triangleq \left(diag\left\{\left[\alpha_1(k) \;\; \alpha_2(k) \;\; \cdots \;\; \alpha_N(k)\right]\right\}\right)^{-1/2}, \qquad (5)$$

$$\alpha_m(k) \triangleq \beta + \frac{1-\beta}{|\boldsymbol{\Omega}_\mathbf{A}| \cdot D_m(k)} \sum_{i \in \boldsymbol{\Omega}_\mathbf{A}} D_m(k) - d_m(k,i), \qquad (6)$$

and $d_m(k,i) \triangleq \left\|\tilde{\mathbf{v}}_i[k] - \hat{\mathbf{s}}_m[k-1]\right\|$ is the distance between the m^{th} feature of and the i^{th} observable motion vector originating from a particular moving region. Here, $D_m(k) \triangleq \max_i\{d_m(k,i)\}$, indicates the maximum distance between a particular feature and the observable motion vectors used to produce its temporal correspondence. In (6), we let β equal 1, ½, and 0 for $\hat{\mathbf{s}}_m[k-1] \in \mathbf{\Omega_A}$, $\hat{\mathbf{s}}_m[k-1] \in \mathbf{\Omega_B}$, and $\hat{\mathbf{s}}_m[k-1] \in \mathbf{\Omega_C}$, respectively. The observations are indicated by

$$\mathbf{x}_m[k] = \left\{ \left|\mathbf{\Omega_A}\right| \cdot D_m(k) - \sum_{i\in\mathbf{\Omega_A}} d_m(k,i) \right\}^{-1} \cdot \\ \cdot \sum_{i\in\mathbf{\Omega_A}} \left(\hat{\mathbf{s}}_m[k-1] + \mathbf{v}_i[k]\right)\left(D_m(k) - d_m(k,i)\right), \quad (7)$$

where $d_m(k,i)$, $\mathbf{v}_i[k]$, and $\hat{\mathbf{v}}_i[k]$ are illustrated in Figure 2 more clearly.

The remaining steps in the filtering procedure mirror those of the standard Kalman filtering state estimate and noise covariance update equations. Thus, they are omitted in this treatment.

2.3. Spatial integration

The spatial integration is performed for each feature point independently. The following treatment addresses spatial integration for the m^{th} feature point, although the index m has been omitted for ease of notation. We introduce random variables, $\xi_j[k] \triangleq \mathbf{s}[k]\big|_j + \mathbf{n}_s[k]\big|_j$ and $\boldsymbol{\lambda}[k] \triangleq \mathbf{y}[k] + \mathbf{n}_y[k]$, where $\mathbf{n}_s[k]\big|_j$ represents some zero-mean distribution of estimation error on the imaging plane corresponding to the j^{th} view. Similarly, $\mathbf{n}_y[k]$ characterizes the zero-mean 3-D reconstruction error inherent in $\boldsymbol{\lambda}[k]$. Let $\xi_j[k]$, $j \in [1, J]$ indicate a family of random variables defined by the vector of probability density functions (PDF), dropping the dependence of all variables on k for simplicity,

$$\mathsf{P}_\Xi\left[\xi_j\right] = (2\pi)^{-N} \left|\mathbf{M}\big|_j\right|^{-\frac{1}{2}} \cdot \\ \cdot \exp\left[-\tfrac{1}{2}\left(\xi_j - \hat{\mathbf{s}}\big|_j\right)^T \mathbf{M}^{-1}\big|_j \left(\xi_j - \hat{\mathbf{s}}\big|_j\right)\right]. \quad (8)$$

where $\mathbf{M}\big|_j$ *contains* the confidence of the m^{th} component of the estimate on the j^{th} view. Under the assumption that the m^{th} feature may be occluded in some views, the algorithm uses $1 < K \le J$ views of the scene to reconstruct an estimate of $\mathbf{y}[k]$. We indicate the set of random variables used in reconstruction by

$$\xi[k;K] \triangleq \left\{\xi_{j_1}[k], \xi_{j_2}[k], \cdots, \xi_{j_K}[k]\right\} \subseteq \left[\xi_1[k], \xi_J[k]\right], \quad (9)$$

where $1 \le j_n \le J$ indicates one of K views upon which $\mathbf{y}[k]$ might be dependent. Again, for the sake of notational simplicity, we drop the explicit dependence of these variables on the frame number, k.

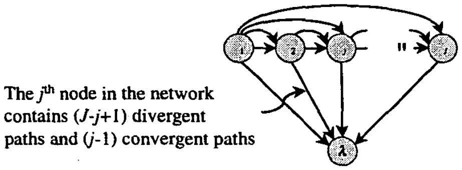
The proposed Bayesian network obtains an estimate of $\mathbf{y}[k]$ and the proper subset $\xi[k;K]$ by calculating

$$\left(\hat{\mathbf{y}}, \hat{\xi}\right) = \arg\max_{\boldsymbol{\lambda},\xi} \left\{\mathsf{P}_{\Lambda,\Xi}[\boldsymbol{\lambda},\xi]\right\}, \quad (10)$$

where, according to Bayes' rule,

$$\mathsf{P}_{\Lambda,\Xi}[\boldsymbol{\lambda},\xi] = \mathsf{P}_\Lambda[\boldsymbol{\lambda} \mid \xi_J, \xi_{J-1}, \cdots, \xi_1] \cdot \\ \cdot \mathsf{P}_\Xi[\xi_J \mid \xi_{J-1}, \xi_{J-2}, \cdots, \xi_1] \cdot \cdots \cdot \mathsf{P}_\Xi[\xi_1]. \quad (11)$$

Representing (11) as a causal network, our task is reduced to finding a topology, $\mathbf{T}\{\xi, \boldsymbol{\lambda}\}$, such that $\mathsf{P}_{\Lambda,\Xi}[\boldsymbol{\lambda},\xi]$ is maximized. In (11), $\mathsf{P}_\Lambda[\boldsymbol{\lambda} \mid \xi_J, \xi_{J-1}, \cdots, \xi_1]$ models 3-D reconstruction noise, $\mathsf{P}_\Xi[\xi_j]$ models 2-D observation noise, and the remaining conditional densities model the effects of occlusion and correlation *between* various views of the scene. The generalized topology of (11) is illustrated in Figure 3.



The j^{th} node in the network contains (J-j+1) divergent paths and (j-1) convergent paths

Figure 3. Bayesian network topology.

To determine a topological ordering of the nodes, we must take into consideration the relative significance of various data sources. Using the $\binom{J}{2}$ fundamental matrices for the system, based on the static content associated with each view, we define a probabilistic measure, $0 \le l_{ij} \le 1$. This probability portrays the ratio of points seen from the i^{th} view that have some correspondence in the j^{th} view to the total number of pixels in the i^{th} view. Within a graph-theoretic framework, this quantity states that an inferential dependency exists between these nodes such that one might conjecture that $i \rightarrow j$ with a confidence of l_{ij}. Two identical views of a scene are indicated by $l_{ij} = 1$, while $l_{ij} = 0$ would suggest two views containing no points in common. Note that the converse of the former statement does not necessarily follow. With fixed cameras, such a matrix, $\mathbf{L} = \{l_{ij}\}$, is easily populated with arbitrary levels of precision using automatic, semi-automatic, or even manual assessment. For a system consisting of J cameras, the nodal ordering is specified by

$$\mathbf{T}\{\xi, \boldsymbol{\lambda}\} \equiv \mathbf{T}\{\tau_1, \tau_2, \cdots, \tau_{J+1}\}\big|_{\tau_{J+1}=\boldsymbol{\lambda}}, \quad (12)$$

where the nodes are placed in descending order of confidence from τ_1, which indicates the root of the BBN, to $\boldsymbol{\lambda}$, which has no dependencies. This topological ordering (12) varies for each feature being tracked and, therefore, must also consider the confidence of temporal tracking. For the m^{th} feature in the j_n^{th} view of the scene we have a confidence metric

$$\phi_{mj_n}[k] \triangleq \alpha_m(k)\big|_{j_n} \cdot \left(1 - \tfrac{1}{J}\sum_{i=1}^{J} l_{j_n i}\right), \quad (13)$$

where $\alpha_m(k)$ for an arbitrary view is given in (6). The random variable for the n^{th} node, τ_n, in the network is equal to ξ_{j_n}, where the views of the scene are ranked according to

$$\phi_{mj_1}[k] \geq \phi_{mj_2}[k] \geq \cdots \geq \phi_{mj_K}[k]. \qquad (14)$$

To calculate (10), we must define the three fundamental densities provided in (11). In addition to the *a priori* distribution at each imaging plane, as indicated by (8), we require density functions for $\boldsymbol{\lambda}$ and ξ_{j_β}, each conditioned upon a collection of other views. Let us refer to this collection of random variables as $\tilde{\xi} \triangleq \{\xi_{j_1}, \xi_{j_2}, \cdots, \xi_{j_B}\} \subseteq \xi$ such that $j_\beta \notin [j_1, j_B] \subseteq [j_1, j_K]$. We introduce two transformation functions, $\mathsf{B}(\cdot)$ and $\mathsf{H}_j(\cdot)$, where the first function maps a collection of random variables, $\tilde{\xi}$, from 2-D to 3-D, while the latter transformation projects a random variable, $\boldsymbol{\lambda}$, from 3-D to the imaging plane of the j^{th} camera. These functions are analogous to those used in error propagation within the 3-D reconstruction literature [21]. Due to the inherent complexity of these transformations, however, it is not always possible to provide a simple, analytical representation of the noise propagation.

To estimate the transformed distributions, we first assume a noise model for the reconstructed and projected density functions. For density functions transformed by $\mathsf{B}(\cdot)$, we assume a 3-D Gaussian distribution with $\mathsf{N}(\boldsymbol{\lambda}', \mathbf{R})$, while for those functions transformed by $\mathsf{H}_j(\cdot)$, we assume a 2-D distribution of $\mathsf{N}(\xi_j', \mathbf{U}_j)$. These distributions are calculated using the notion of random sampling. For $\mathsf{N}(\boldsymbol{\lambda}', \mathbf{R})$, we choose one point at random (distributed according to ξ_{j_1}, ξ_{j_2}, etc...) on each of the imaging planes corresponding to $\tilde{\xi}$. These B points are then triangulated in the standard way [22] to form a random observation in 3-D Cartesian space. We continue this sampling process until enough observations exist to estimate $\boldsymbol{\lambda}'$ and $\mathbf{R}$ with sufficient confidence. A similar sampling procedure is used to estimate ξ_j' and $\mathbf{U}_j$, where points are selected at random in 3-D according to $\mathsf{N}(\boldsymbol{\lambda}', \mathbf{R})$, which also uses random samples from the j^{th} view, and then projected onto the j^{th} imaging plane. The random 3-D positions will project to a random distribution on the imaging plane that provides a set of observations for estimating ξ_j' and $\mathbf{U}_j$. The transformations, $\mathsf{F}(\cdot)$ and $\mathsf{G}(\cdot)$, for an arbitrary imaging plane collectively perform the same mathematical function as $\mathsf{H}_j(\cdot)$.

Starting with the two most confident views of the scene, corresponding to τ_1 and τ_2 in the network, we construct $\mathsf{P}_{\Lambda,\Xi}[\boldsymbol{\lambda}, \xi]$ under the assumption that a conditional independence exists

$$\mathsf{P}_{\Lambda,\Xi}[\boldsymbol{\lambda}, \xi] \equiv \mathsf{P}_{\Lambda}[\boldsymbol{\lambda} \mid \xi] \cdot \mathsf{P}_{\Xi}[\xi] = \mathsf{P}_{\Lambda}\left[\boldsymbol{\lambda} \mid \tilde{\xi}\right] \cdot \mathsf{P}_{\Xi}\left[\tilde{\xi}\right], \quad (15)$$

where $\tilde{\xi} = \{\xi_{j_1}, \xi_{j_2}\}$. To calculate (15), we introduce

$$\begin{aligned} \mathsf{P}_{\Lambda}\left[\boldsymbol{\lambda} \mid \tilde{\xi}\right] &= (2\pi)^{-\frac{3N}{2}} |\mathbf{R}|^{-\frac{1}{2}} \cdot \\ &\quad \cdot \exp\left[-\tfrac{1}{2}[\boldsymbol{\lambda} - \boldsymbol{\lambda}']^T \mathbf{R}^{-1} [\boldsymbol{\lambda} - \boldsymbol{\lambda}']\right] \end{aligned} \qquad (16)$$

and

$$\begin{aligned} \mathsf{P}_{\Xi}\left[\xi_{j_\beta} \mid \tilde{\xi}\right] &= (2\pi)^{-N} \left|\mathbf{U}_{j_\beta}\right|^{-\frac{1}{2}} \cdot \\ &\quad \cdot \exp\left[-\tfrac{1}{2}\left[\xi_{j_\beta} - \xi_{j_\beta}'\right]^T \mathbf{U}_{j_\beta}^{-1} \left[\xi_{j_\beta} - \xi_{j_\beta}'\right]\right] \end{aligned} . \qquad (17)$$

The maximization of (15) is trivial due to the normal distribution of each variable. The solution parallels that of a weighted least-squares problem, where $\tilde{\xi}$ are the observations and the *a priori* density in (8) is analogous to the weighting factor for a particular observation. We iteratively modify the topology of the graph, adding nodes of successively lower confidence, until the dynamic topological ordering satisfies (10). The resulting estimate of 3-D position, $\hat{\mathbf{y}}[k]$, and corresponding noise covariance, $\mathbf{R}[k]$, are the input to a Kalman filter that encourages a 3-D trajectory with temporal continuity.

2.4. Temporal integration

The state-based feature vector represents both the location and velocity of points in 3-D and is indicated by

$$\boldsymbol{\sigma}[k] = \left[\boldsymbol{\sigma}_1[k] \;\; \boldsymbol{\sigma}_2[k] \;\; \cdots \;\; \boldsymbol{\sigma}_{(2 \cdot N)}[k]\right]^T, \qquad (18)$$

where $\boldsymbol{\sigma}_m[k]$, $m \leq N$ indicates the ideal 3-D position of the m^{th} feature within the k^{th} frame and

$$\boldsymbol{\sigma}_m[k] = \left.\frac{\partial \boldsymbol{\sigma}_b[k]}{\partial k}\right|_{b=m-N, m>N}. \qquad (19)$$

An estimate of the 3-D feature vector in (18) is denoted by (20) where we use a dynamic model for $\boldsymbol{\Psi}[k]$ with constant velocity and linear 3-D displacement such that

$$\begin{aligned} \hat{\boldsymbol{\sigma}}[k \mid k-1] &= \boldsymbol{\Psi}[k]\hat{\boldsymbol{\sigma}}[k-1 \mid k-1] \\ &= [\gamma_1[k] \; \gamma_2[k] \cdots \gamma_N[k] \; 1 \; 1 \cdots 1]^T \end{aligned}, \qquad (20)$$

where

$$\gamma_m[k] = 2\hat{\boldsymbol{\sigma}}_m[k-1 \mid k-1] - \hat{\boldsymbol{\sigma}}_m[k-2 \mid k-2]. \qquad (21)$$

We develop an error covariance matrix, $\boldsymbol{\Gamma}[k \mid k-1]$, that depicts our confidence in the predictions of the state estimates. The update equation is indicated by

$$\boldsymbol{\Gamma}[k \mid k-1] = \boldsymbol{\Psi}[k]\boldsymbol{\Gamma}[k-1 \mid k-1]\boldsymbol{\Psi}^T[k] + \mathbf{Q}[k], \qquad (22)$$

where $\mathbf{Q}[k]$ represents a Gaussian noise covariance matrix which is iteratively modified over time to account for the deviations between the predictions and corrections of the state estimates. The system then constructs a Kalman gain matrix, $\mathbf{D}[k]$, according to

$$\mathbf{D}[k] = \boldsymbol{\Gamma}[k \mid k-1]\boldsymbol{\Phi}^T[k]\mathbf{Y}^{-1}[k], \qquad (23)$$

where

$$\mathbf{Y}[k] = \tfrac{1}{2}\{\mathbf{R}[k] + \boldsymbol{\Theta}[k]\} + \boldsymbol{\Phi}[k]\boldsymbol{\Gamma}[k \mid k-1]\boldsymbol{\Phi}^T[k] \qquad (24)$$

and $\boldsymbol{\Phi}[k] = [\mathbf{I} \,\vdots\, \mathbf{0}]_{N \times 2N}$ indicates the linear observation matrix and $\boldsymbol{\Theta}[k]$ is a recursively updated observation noise covariance matrix. The remaining steps in the filtering procedure mirror those of the standard Kalman filtering state estimate and noise covariance update equations. Thus, they are omitted in this treatment.

3. Experimental results

To test the proposed contribution, we use 600 frames of synchronized video data taken from three distinct views of a home environment. The underlying scene captures an informal social gathering of four people, each of whom is characterized by five semantic features that are selected at first sight and tracked throughout the remainder of the sequence. For features, we use the top of each shoe, the transition between the sleeve and the arm, and the top of the torso underneath the neck as seen from the front of the body. If a desired feature is initially not visible due to self-occlusion, but its position in one or more views can be estimated with fair accuracy, it is labeled and tracked as an occluded point until it becomes visible.

To demonstrate the success of the proposed tracking algorithm, we show in Figure 4 trajectories for each of the moving people in the scene. Each row shows a five second interval for a particular individual over all three views.

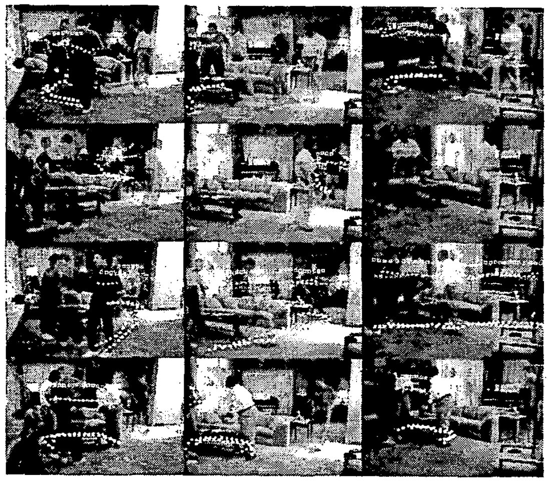


Figure 4. Trajectories of various features.

To quantify the accuracy of the proposed tracking system, we calculate the average absolute error between the automatically generated feature locations and the corresponding ground-truth data (annotated by hand). When 3-D data is available, we characterize the tracking error, ε_B, by using the absolute difference between the ground-truth and the 3-D feature projection at each imaging plane. The reported error is taken on the imaging plane carrying the *maximum* absolute difference over all J views. If a feature can only be tracked from one view, however, we simply calculate the absolute error between the ground-truth and the tracking results of monocular image sequence processing.

For exhaustively quantifying the data, then, we must develop a baseline for three views of four people, each with five features, over 600 frames. As this would clearly be no less than an overwhelming task, we choose to faithfully represent the entire sequence using a random sample of only 60 frames. The fundamental hypothesis of the proposed contribution is based upon an assumed correlation between tracking accuracy and various configurations of visible and occluded features. As is such, we group features into any of B bins, where for a total of J views and F confidence levels, B is the number of unique solutions to

$$\sum_{i=1}^{F} O_i = J, \text{ where } 0 \le O_i \le J, \ \{F, O_i, J\} \in \mathbb{N}, \quad (25)$$

and O_i is the number of views with the i^{th} confidence level. Using the metric in (6), the algorithm associates a confidence level for each feature. We quantize the number of levels to three, where a confidence level of *0* (L_V) suggests that the feature is being tracked with high accuracy, a level of *1* (L_O) indicates that the feature is being tracked with relatively low accuracy, and a level of *2* (L_M) identifies a feature that is no longer in the field of view. Using these three levels, it can be shown that we have $\frac{1}{2}J^2 + \frac{3}{2}J + 1$ bins for any J combination of cameras. Each bin is represented using a triplet, where for any feature the first number, O_V, indicates the number of cameras that presumably have an unobstructed view, the second number, O_O, represents the number of cameras that are thought to have an occluded view, and the third number, O_M, specifies the number of cameras for which the feature is outside the field of view.

Figure *5* summarizes the tracking error for the proposed Bayesian technique (ε_B) while comparing to the performance of a more simplistic averaging approach (ε_A). For the latter method, multiple observations are combined in the least-squares sense in order to combat the effects of observation noise. We draw the reader's attention to a number of important characteristics in Figure 5. Each bin in the histogram represents a different class of features, denoted by a triplet, where the first, second, and third digits indicate the number of cameras tracking a given feature with high, low, and no accuracy, respectively. The labeled percentages indicate the percent increase in error of one approach over another for each class of features. As indicated by the data, using the proposed BBN for data fusion produces lower tracking errors for any class of features for which more than two observations exist (i.e., bins 300, 210, 120, and 030). As demonstrated by the transitions from bins 111 to 120/210 and 201 to 210/300 in Figure 5, we only witness a substantial decrease in error when the third observation has a relatively high confidence. In contrast, the Bayesian network effectively considers only the most likely observations.

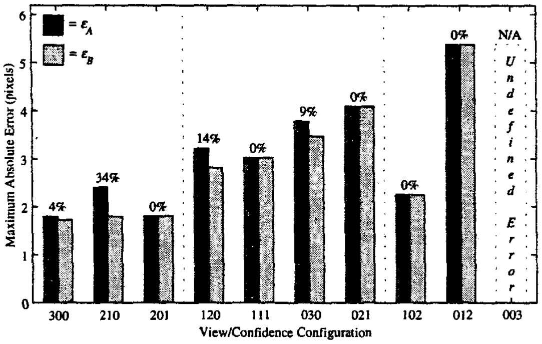


Figure 5. Summary of tracking results.

If we consider our specific distribution of features as a function of configuration over the entire sequence, as indicated in Figure 5, the difference in error between ε_A and ε_B is as low as 4%, as high as 34%, and 14.5% on average. As further indicated by Figure 6, the total number of features for which the proposed algorithm improves the tracking accuracy accounts for 42% of all features.

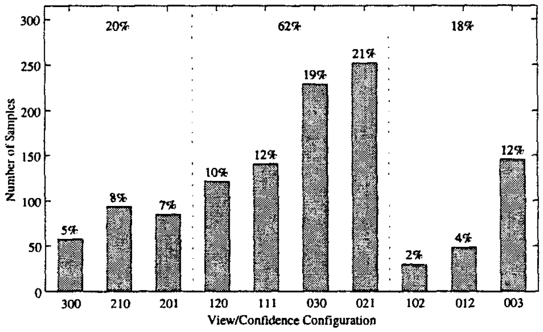


Figure 6. Distribution of features.

We further quantify the performance of the individual system components by defining a metric, χ, that measures the ratio of the feature position error at the output of a particular component to the error of the prediction at the input of the predictor-corrector filter. Here, the error is measured as the Euclidean distance between a calculated feature position and the known ground truth. Ideally, this ratio will decrease monotonically as the data traverses the flow of the system, thus indicating that each component marginally improves the overall tracking accuracy. We report the mean value of this metric over all views of the scene at the output of the predictor-corrector filter ($\bar{\chi}_{PC}$), the Bayesian network ($\bar{\chi}_{BN}$), and the 3-D Kalman filter ($\bar{\chi}_{KF}$) in Table 1. The results only consider those configurations for which a feature is within the field of view for two or more cameras.

All major sources of error in the tracking results can be traced back to occlusion. A careful examination of the sequence indicates that self-occlusion is often a stronger culprit than other forms, such as occlusion due to multiple object interactions and scene clutter. Because we use the less computational, yet more simplistic, approach of model-free tracking, the motion of features in self-occlusion is often predicted using erroneous vector estimates within the foreground of the moving person in question. The result is a trajectory that is correct at a global scale, but corrupt at finer resolutions.

Table 1. System component tracking errors.

	View/Confidence Configuration						
	300	210	201	120	111	030	021
$\bar{\chi}_{PC}$	0.18	0.58	0.21	0.78	0.66	0.86	0.87
$\bar{\chi}_{BN}$	0.09	0.13	0.15	0.36	0.34	0.36	0.43
$\bar{\chi}_{KF}$	0.10	0.13	0.12	0.21	0.23	0.32	0.38

One solution to the problem of self-occlusion might be the introduction of an *a priori* motion model. However, given the proposed framework for the fusion of video data from multiple cameras, a simpler (and provably more optimal) solution might be to extend the number of views, thus migrating towards the increasingly popular notion of next generation ubiquitous computing. To develop a full appreciation of the proposed technique, we invite and encourage the reader to visit our website to inspect the multi-view tracking results in their entirety.

4. Conclusions

We introduce a novel technique for tracking interacting human motion using multiple layers of temporal filtering coupled by a simple Bayesian belief network for multiple camera fusion. The system uses a distributed computing platform to maintain real-time performance and multiple sources of video data, each capturing a distinct view of some scene. To maximize the efficiency of distributed computation each view of the scene is processed independently using a dedicated processor. The processing for each view is based on a predictor-corrector filter with Kalman-like state propagation that uses measures of state visibility and sparse estimates of image motion to produce observations.

The corrected output of each predictor-corrector filter provides a vector observation for a Bayesian belief network. The network is characterized by a dynamic, multidimensional topology that varies as a function of scene content and feature confidence. The algorithm calculates an appropriate input configuration by iteratively resolving independency relationships and *a priori* confidence levels within the graph. The output of the network is a vector of 3-D positional data with a corresponding vector of noise covariance matrices; this information is provided as input to a standard Kalman filtering mechanism. The proposed method of data fusion is compared to the more basic approach of data averaging. Our results indicate that for any input configuration consisting of more than two observations per feature the method of Bayesian fusion is superior.

Acknowledgments

This research was made possible by generous grants from the Center for Future Health, the Keck Foundation, and Eastman Kodak Company. We would also like to acknowledge Terrance Jones for his assistance and organizational efforts in providing synchronized multi-view sequences of multiple people in action.

References

[1] T. Boult, R. Micheals, A. Erkan, P. Lewis, C. Powers, C. Qian, and W. Yin, "Frame-rate multi-body tracking for surveillance," *Proc. of the DARPA Image Understanding Workshop*, Monterey, CA, 20-23 November 1998, pp. 305-313.

[2] S. L. Dockstader and A. M. Tekalp, "Tracking multiple objects in the presence of articulated and occluded motion," *Proc. of the Workshop on Human Motion*, Austin, TX, 7-8 December 2000, pp. 88-95.

[3] A. D. Wilson and A. F. Bobick, "Parametric Hidden Markov Models for Gesture Recognition," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 21, no. 9, pp. 884-890, September 1999.

[4] A. Pentland, "Looking at People: Sensing for Ubiquitous and Wearable Computing," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 107-119, January 2000.

[5] J. M. Rehg and T. Kanade, "Model-based tracking of self-occluding articulated objects," *Proc. of the Int. Conf. on Computer Vision*, Cambridge, MA, 20-23 June 1995, pp. 618-623.

[6] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland, "Pfinder: Real-Time Tracking of the Human Body," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 780-785, July 1997.

[7] S. Wachter and H.-H. Nagel, "Tracking Persons in Monocular Image Sequences," *Computer Vision and Image Understanding*, vol. 74, no. 3, June 1999.

[8] Y.-S. Yao and R. Chellappa, "Tracking a Dynamic Set of Feature Points," *IEEE Trans. on Image Processing*, vol. 4, no. 10, pp. 1382-1395, October 1995.

[9] M. Isard and A. Blake, "Condensation - Conditional Density Propagation for Visual Tracking," *Int. J. of Computer Vision*, vol. 29, no. 1, pp. 5-28, August 1998.

[10] D.-S. Jang and H.-I. Choi, "Active Models for Tracking Moving Objects," *Pattern Recognition*, vol. 33, no. 7, pp. 1135-1146, July 2000.

[11] J. MacCormick and A. Blake, "A probabilistic exclusion principle for tracking multiple objects," *Proc. of the Int. Conf. on Computer Vision*, Kerkyra, Greece, 20-27 September 1999, pp. 572-578.

[12] S. J. McKenna, S. Jabri, Z. Duric, and H. Wechsler, "Tracking interacting people," *Proc. of the Int. Conf. on Automatic Face and Gesture Recognition*, Grenoble, France, 28-30 March 2000, pp. 348-353.

[13] I. Haritaoglu, D. Harwood, and L. S. Davis, "W^4: Real-Time Surveillance of People and Their Activities," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 809-830, August 2000.

[14] D. M. Gavrila and L. S. Davis, "3-D model-based tracking of humans in action: a multi-view approach," *Proc. of the Conf. on Computer Vision and Pattern Recognition*, San Francisco, CA, 18-20 June 1996, pp. 73-80.

[15] T. Darrell, G. Gordon, M. Harville, and J. Woodfill, "Integrated Person Tracking Using Stereo, Color, and Pattern Detection," *Int. J. of Computer Vision*, vol. 37, no. 2, pp. 175-185, June 2000.

[16] S. Stillman, R. Tanawongsuwan, and I. Essa, "A system for tracking and recognizing multiple people with multiple cameras," *Proc. of the Int. Conf. on Audio and Video-Based Biometric Person Authentication*, Washington, DC, 22-23 March 1999, pp. 96-101.

[17] A. Utsumi, H. Mori, J. Ohya, and M. Yachida, "Multiple-human tracking using multiple cameras," *Proc. of the Int. Conf. on Automatic Face and Gesture Recognition*, Nara, Japan, 14-16 April 1998, pp. 498-503.

[18] Q. Cai and J. K. Aggarwal, "Tracking Human Motion in Structured Environments Using a Distributed-Camera System," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 21, no. 11, pp. 1241-1247, November 1999.

[19] R. T. Collins, A. J. Lipton, and T. Kanade, "Introduction to the Special Section on Video Surveillance," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 745-746, August 2000.

[20] J. L. Barron, D. J. Fleet, and S. S. Beauchemin, "Systems and Experiment: Performance of Optical Flow Techniques," *Int. J. of Computer Vision*, vol. 12, no. 1, pp. 43-77, 1994.

[21] Z. Sun, *Object-Based Video Processing with Depth*, Ph.D. Thesis, University of Rochester, 2000.

[22] E. Trucco and A. Verri, *Introductory Techniques for 3-D Computer Vision*, Upper Saddle River, NJ: Prentice-Hall, 1998.