2008 ACM/IEEE International Conference on Distributed Smart Cameras

A NOVEL EVIDENCE ACCUMULATION FRAMEWORK FOR ROBUST MULTI-CAMERA PERSON DETECTION

Hidekazu Iwaki, Gaurav Srivastava, Akio Kosaka, Johnny Park and Avinash Kak

Robot Vision Lab, Electrical and Computer Engineering
Purdue University West Lafayette IN 47907
{hiwaki, gsrivast, kosaka, jpark, kak}@purdue.edu

ABSTRACT

We propose a novel evidence accumulation framework that accurately estimates the positions of humans in a 3D environment. The framework consists of a network of distributed agents having different functionalities. The modular structure of the network allows scalability to large surveillance areas and robust operation. The framework does not assume reliable measurements in single cameras (referred to as ’sensing agents’ in our framework) or reliable communication between different agents. There is a position uncertainty associated with single camera measurements and it is reduced through an uncertainty reducing transform that performs evidence accumulation using multiple camera measurements. Our framework has the advantage that single camera measurements do not need to be temporally synchronized to perform evidence accumulation. The system has been tested for detecting single and multiple humans in the environment. We conducted experiments to evaluate the localization accuracy of the position estimates obtained from the system by comparing them with the ground truth. Also, two different configurations of the agents were tested to compare their detection performance.

Index Terms— camera networks, distributed processing, evidence accumulation, uncertainty reduction.

I. INTRODUCTION

The biggest advantage of multi-camera surveillance networks over single camera systems is their ability to combine information from different cameras into scene-level representations that yield enhanced awareness of the monitored environment . But this ability depends critically on how the information is combined from the different cameras. We obviously need an evidence accumulation framework that is well-principled with regard to combining the uncertainties in the information gleaned from each camera. We also want such a framework to scale up easily as more and more cameras are added to the network. As a camera network becomes large, it is extremely difficult to synchronize image capture by the different cameras. Therefore, we would want the framework to combine information from the different cameras taking into account the uncertainty in image acquisition times. The goal of this paper is to present such an evidence accumulation framework.

Our proposed framework consists of a hierarchy of agents. The lowest level of this hierarchy consists of ’sensing agents’; they extract candidate shapes and features. Higher levels of the agent hierarchy deal with: 1) the local accumulation of supporting evidence for the shape/feature hypotheses that are output by the sensing agents; and 2) the aggregation of the hypotheses at a more global level. Note that the candidate shapes/features that are output by the lowest level of the agent hierarchy suffer from high false-positive rates because of complex backgrounds, occlusions, rather limited fields of view of the individual cameras, and so on (Figure 1). It is the accumulation of evidence at the higher levels of the hierarchy that progressively eliminates the false positives and provides accurate estimation of the human positions in the monitored environment.

II. RELATED WORK

Many evidence accumulation schemes have been proposed in the multi-camera visual surveillance literature. These include the schemes reported in [1], [2], [3], [4], [5], [6], [7], [8], [9] and others. In [1] and [10], a person’s 3D location is estimated by triangulation of 3D rays directed along the line joining camera focal points and the person’s centroid in 2D image planes. A pseudo-intersection point is computed that minimizes the sum of the squared distance to each pointing ray. Bayesian networks have also been used for multi-camera evidence accumulation [4], [11], [12]. In [11], a Bayesian belief network is used to match subjects across different cameras by integrating geometry- and recognition-based modalities, whereas, in [4], the Bayesian net fuses independent observations from multiple cameras by iteratively resolving independency relationships and confidence levels within the net. The system described in [7], [8] is based on FOV (field of view) lines of the cameras to establish correspondences between the views of the same object as seen in different uncalibrated cameras. In [13], image-based sensor observations are associated with scene-based object hypotheses using features that are viewpoint independent e.g. object location, object class (human, vehicle etc.) and

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

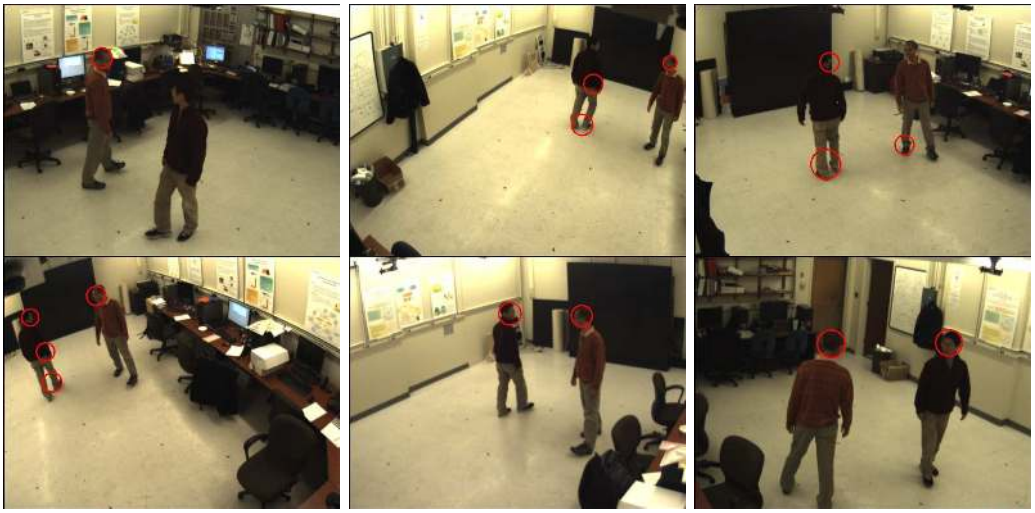


Fig. 1. Images from a multi-camera test sequence with complex background. They were acquired at approximately the same time. Red circles depict the detected head candidates (both true heads and false positives). Inspite of a large number of false positives due to the complex background, the proposed evidence accumulation scheme generates accurate 3D head positions. A demo video of our multi camera person detection system can be viewed at `http://cobweb.ecn.purdue.edu/RVL/movies/MultiCameraPersonDetection_ICDSC08.wmv`

color. The associations are made based on comparing the features of a new observation against the features stored for each existing object hypothesis using a match score function. The work in [14] describes a region-based stereo algorithm capable of finding 3D points inside an object knowing only the projections of the object (as a whole) in two views. The work reported in [15] addresses the problem of selecting the best camera position for extracting the desired human motion information. The human position, body orientation and body-side estimation is performed by determining the camera viewpoints where these features can be easily estimated and maximizing the joint probabilities of observations obtained from multiple cameras.

The work we report here carries out evidence accumulation with a framework of agents possessing heterogeneous characteristics. These agents cooperate to estimate 3D human positions in real time, followed by determination and visualization of their trajectories. Due to the modular agent-based processing architecture, the proposed framework is well suited to large-scale surveillance applications since new agents can be integrated seamlessly. The evidence accumulation scheme works well even when the different cameras are not synchronized with regard to their image acquisition times. The unsynchronized observations allow for denser temporal observations of the 3d environment and avoid redundancy among multiple observations when a large number of cameras is deployed in the network [16].

The paper is organized as follows. Section 2 describes the problem of 3D position estimation that we try to solve and the assumptions about the environment and our current setup. Section 3 details the agent based architecture which achieves multiple functionalities through cooperative interaction between heterogeneous agents. Section 4 explains the single sensor processing of acquired images, while section 5 elaborates the novel evidence accumulation scheme to obtain accurate 3D position estimates. Experimental results are presented and discussed in section 6. Section 7 gives some concluding remarks followed by possible future extensions of the current work. Finally, section 8 acknowledges the funding support for this research.

III. PROBLEM DESCRIPTION

Our overall goal is to develop a cooperative processing architecture for detecting and tracking multiple humans in an environment and visualizing their trajectories. The work presented in this paper solves a sub-problem of the human tracking problem: first detect the humans in the environment and estimate their positions using the individual cameras in the network, and then combine the information gleaned from the individual cameras to achieve higher localization accuracy of the estimated positions and the reduction of false detections. Our setup consists of 12 cameras monitoring an indoor rectangular area $(8m \times 5m)$. The cameras are grouped into 4 clusters as shown in Figure 2. In solving the detection and localization problem, we have made the following assumptions:

- The environment is defined in terms of the world coordinate frame that is taken to be the reference coordinate frame.
- All cameras are calibrated with respect to the world coordinate frame.
- Image capture by the different cameras is not synchronized and the images are acquired with time stamp information.

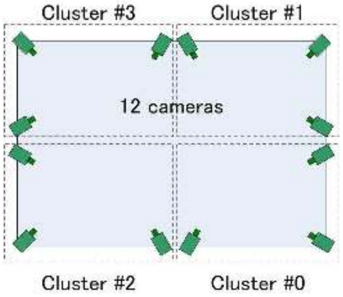


Fig. 2. **Camera configuration used for the evidence accumulation framework results reported in this paper:** There are 12 cameras grouped into 4 clusters, each monitoring a small part of a rectangular area.

- The cameras are connected to PCs that perform all image processing operations.
- The PCs can communicate with one another via either wired or wireless network connections.
- Multiple humans may exist in the environment viewed by the network of cameras.
- In this paper, the position of a human is represented by the human head position.

IV. AGENT BASED ARCHITECTURE

The cooperative processing architecture consists of the following agents:

- Sensing Agent
- Cluster Leader Agent
- Monitoring Agent
- Visualization Agent

These agents are software processes running on PCs that are connected by wired or wireless network links. Multiple such agents may run on a single PC. The agents may also control hardware such as cameras for image capture or display devices for visualizing the trajectories of the detected humans. Figure 3(a) shows a generic view of our agent based architecture and Figure 3(b) shows an example implementation that was used for the results reported here.

IV-A. Sensing Agent

The sensing agents are situated at the bottom of the hierarchy of agents. Ideally, in distributed sensor networks, a sensor node consists of a sensor, a processing module, and a communication module. In our current setup, we simulate a sensor node by a sensing agent. It is a software agent running on a PC, that utilizes an IEEE 1394 firewire camera for image capture, performs local processing on the acquired images and sends some data to other agents (specifically the cluster leader) at the next higher level of the agent hierarchy. As mentioned earlier, the images are captured with time-stamp information. Local processing involves the application of a background subtraction algorithm [17] to obtain contours of foreground objects and extracting human-head like object regions from these contours. The extracted head-like regions are also called *head region candidates*. Head extraction algorithm is explained in section V. Sensing always involves false detections. So a sensing agent is not expected to always successfully detect the human heads. Its responsibility is only to detect the head region candidates. In the ensuing discussion, we will refer to single camera head region candidates as ’measurements’. The sensing agent then sends a message including the measurements to a cluster leader. It is worth mentioning that actual images are not sent, rather only small datasets are sent. This is shown in Figure 4.

IV-B. Cluster Leader

A cluster leader implements our evidence analysis and accumulation algorithm to unify the information received from lower level agents in the agent hierarchy. Note that the hierarchical organization of the agents shown in Figure 3(a) allows for the node below a cluster leader to be either a sensing agent or another cluster leader agent. A cluster leader agent receives messages containing measurements or position estimates from lower level agents and uses them to accumulate evidence for accurate 3D position estimation. There are two types of position estimates:

1) Candidate position estimate (CPE): This is generated by a lower-level cluster leader as a result of integrating one or more measurements received from different sensing agents. It is represented by $(\overline{\mathbf{p}}, S)$, where $\overline{\mathbf{p}}$ is the mean vector representing the candidate position and S is the covariance matrix representing position uncertainty. $\overline{\mathbf{p}}$ and S are specified in 3D world coordinates.
2) Validated position estimate (VPE): When a CPE is able to accumulate evidence from 3 or more measurements, it is said to be validated and is then known as a validated position estimate (VPE). The integration and validation of position estimates is performed using Mahalanobis distances and weighted recursive least squares technique [18]. A VPE is also denoted by $(\overline{\mathbf{p}}, S)$. A CPE may or may not represent an actual human head depending on how many measurements are integrated into it but a VPE represents the position of an actual human head.

Once the VPEs are generated, “unnecessary” measurements are eliminated within the cluster leader to avoid data redundancy and to ensure that each measurement is associated with a unique VPE. The cluster leader then sends a message to a higher-level cluster leader containing the VPEs and also the CPEs that it could not validate. If a cluster leader at the topmost level can not validate any of the CPEs, they are discarded. A top-level cluster leader sends all the VPEs to the monitoring agent and the visualization agent for generating the trajectories and for the visualization of the detected human heads.

IV-C. Monitoring and Visualization Agents

Since the current paper focuses primarily on accurate head position detection using evidence accumulation, we fill focus on the functions of the sensing agents and the cluster leader agents in the following sections. The monitoring and the visualization agents will be presented in detail in future publications. Suffice here to say that the monitoring agent is responsible for monitoring the object/humans found in the environment by associating tracking labels with such objects and the visualization agent provides a user interface for visualizing the 3D environment along with the objects/humans found in it.

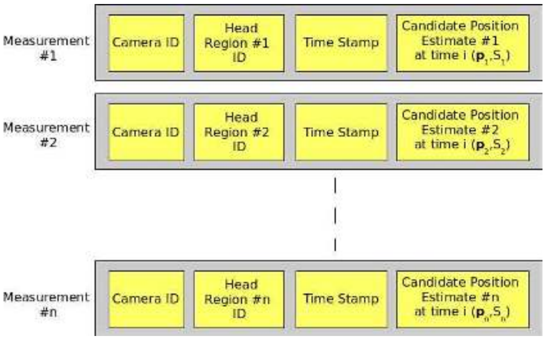


Fig. 4. **Data transmitted from a sensing agent to the cluster leader:** At each time instant i, a data record containing the position estimates for all the detected head region candidates along with the associated timestamp information is sent. No images are sent.

IV-D. Connectivity and Communication Issues

In a distributed network (wired or wireless), reliability or lack thereof is an important issue. We do not wish to assume a reliable network and we want our framework to allow for fault conditions such as some sensor nodes going down or some communication links failing during a detection and tracking task. To realize an unreliable network, we use the UDP messaging protocol rather than the TCP/IP protocol. A cluster leader integrates the information received from the lower level nodes. Therefore, since the system allows for cluster leader failure, the network connections between the sensing agents and the cluster leaders are reconfigurable dynamically. That way, if a cluster leader node fails, the sensing agents connected to it can start sending their data to other active cluster leaders in the network.

IV-E. Configuring the Agent Hierarchy

Depending on the number of sensing agents in the camera network, there may be one or more cluster leaders and they may be arranged in multiple layers of the agent hierarchy. There is a tradeoff involved between the numbers of levels of the hierarchy in the architecture versus the communication delays in the network. On the one hand, the sensing agents and the cluster leaders may be configured in multiple layers as shown in Figure 3(a), so that there are multiple clusters of sensing agents and each cluster's data is processed by one cluster leader. Such a configuration will have higher cumulative communication delays compared to a simple network where all the sensing agents are directly connected to a single cluster leader that does all the integration and validation processing. On the other hand, it is typical of wireless sensor networks that the sensing agent nodes may have limited communication range and so may not be able to send their data to a single cluster leader. Therefore formation of multiple clusters may be necessary.

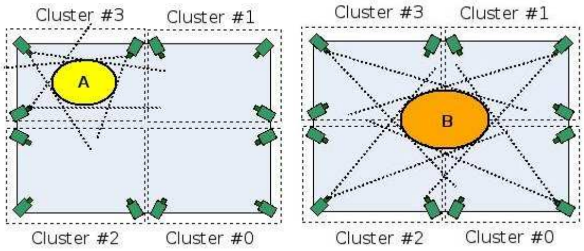
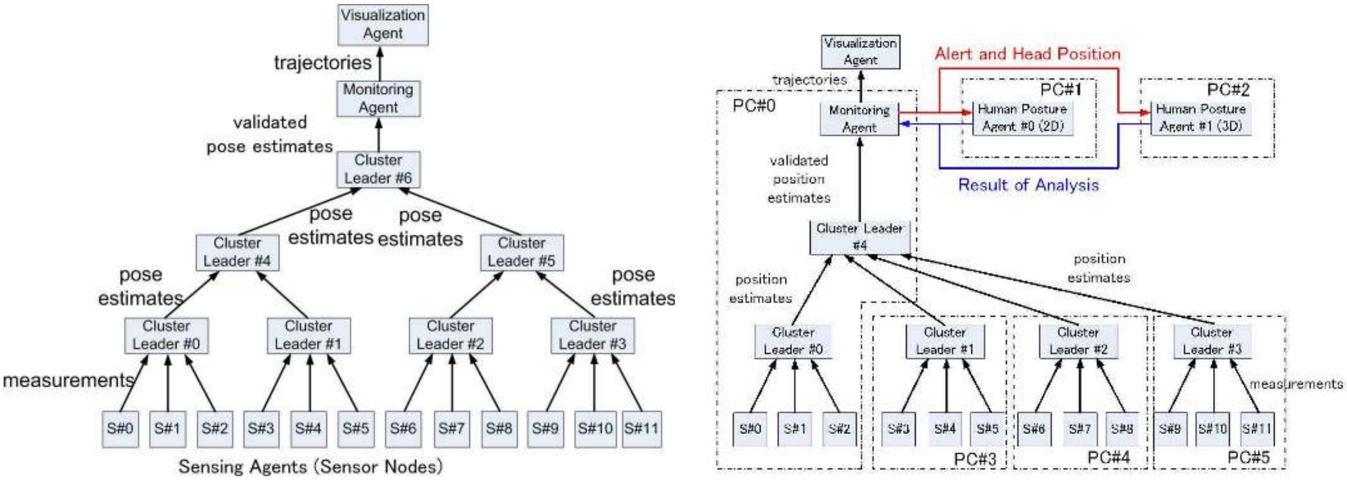


Fig. 5. **Why do we need multiple levels of Cluster Leaders:** Head position validation requires data from at least three sensing agents. Positions in Area A are coverable by three sensing agents from a single cluster; so a single cluster leader can perform validation. But positions in area B are not coverable by three sensing agents of a single cluster. Therefore multiple levels of cluster leaders are needed for integration and validation in area B.

If multiple cluster formation is allowed, each cluster may be able to cover only a portion of the entire monitored area. In our current system implementation, a cluster leader requires measurements from at least three sensing agents to obtain a VPE. As shown in figure 5(a), the cluster leader for cluster #3 can validate all the locations within area A. But area B (figure 5(b)) is not coverable by at least three sensing agents of any one cluster; so no single cluster leader can validate the locations in this area. So the cluster leaders corresponding to all the four clusters need to send their position estimate data to a higher level cluster leader to perform a second level of integration. This scenario justifies the need for having multiple levels of cluster leaders in our architecture.

V. SINGLE VIEW HEAD DETECTION

The human head detection in single camera images involves contour analysis of foreground silhouettes. The algorithm we use for that purpose is based on the work of Zhao [19]. This work deals with shape decomposition and body part identification in line-approximated contours of foreground objects that are assumed to be humans (see Figure 6 for details). Body part identification is followed by the extraction of edge segment boundaries. An edge segment boundary is defined as that contour whose boundary contains



(a) Generic view

(b) Example Implementation

Fig. 3. **An agent-based hierarchical processing architecture for the detection of humans and their localization.**

only one cut; cuts are shown as red lines in figure 6 (c). In figure 6 (d), edge segment boundaries are the boundaries of the red colored patches.

For each edge segment boundary, its similarity to a simple head model is computed. The head model (which is assumed to be a circle) is fitted to each edge segment boundary i using the least squares method and its center $\mathbf{x}_{0i}$ and radius r_i calculated. The average fitting error E_i in this calculation is computed as

$$E_i = \frac{1}{N_i} \sum_{j=1}^{N_i} \left(\|\mathbf{x}_j - \mathbf{x}_{0i}\|_2^2 - r_i^2 \right) \tag{1}$$

where $\mathbf{x}_j$'s are the points on edge segment boundary i and there are N_i of them. The similarity of the edge segment boundary i to the head model is computed as

$$Sim_i = \frac{\sqrt{r_i^2 - E_i}}{r_i} \tag{2}$$

Note that $0 \leq Sim_i \leq 1$. All the edge segment boundaries for which Sim_i exceeds a threshold $(0.6 - 0.8)$ are detected as head region candidates. We assume that an average human head when modeled as a sphere in 3D is approximately 10 inches in diameter. Using this assumption and the estimated radius of the human head region candidate in a single camera image, we can estimate a rough distance d of the human head from the camera, using the relation

$$\frac{d}{F} = \frac{D}{2r} \tag{3}$$

where F is the focal length of camera, r the estimated radius of the head region candidate, and $D = 25.4\ cm(10\ inches)$. Here D is the assumed diameter of the average human head; it is obtained experimentally through measurements on several people. All lengths are assumed in cm units. This equation indicates that r is small for a person far away from a camera (d large) and vice versa. This equation also indicates that $|\triangle d| = (\triangle r/r)\, d$, implying that the uncertainty in d is large for a person far away from a camera and vice versa.

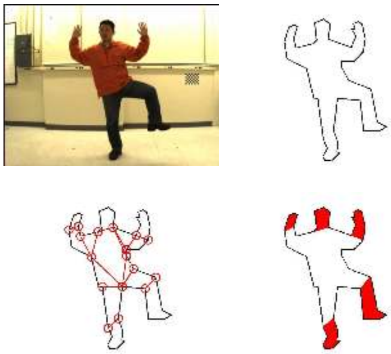


Fig. 6. **Zhao's Shape Decomposition (from [19]):** (a) the original image, (b) line approximated contour of foreground person, (c) computing the negative curvature minima (represented by small circles) and the cuts (represented by red lines) (d) the edge segment boundaries; these are the boundaries of the red colored patches.

Figure 7 presents an idealized representation of the head candidate detected by a single camera for a single human in its field of view. The head candidate is represented in camera coordinate frame by (u, v, d) where (u, v) are the pixel coordinates of the head candidate region mean and d is its distance from the principal center of the camera. The ellipse in the figure represents the uncertainty in d.

The candidate position measurement (u, v, d) obtained from a single camera image is transformed into the world coordinate frame $\mathbf{p} = (x, y, z)$. Since there is always some position uncertainty associated with a camera measurement of the human head position, each measurement is specified by the mean position $\overline{\mathbf{p}}$ and covariance matrix S (see Ap-

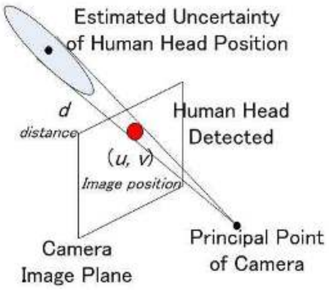


Fig. 7. **Single camera head detection.**

pendix A for mathematical details on how the measurement (u, v, d) is converted to the world coordinate frame).

The measurement from a single camera may not represent the actual position of a human head. That is why we refer to a detected region as a *head candidate* rather than a head. The reason is that certain non-human objects may appear circular in a single camera view and may be mistaken for a human head. Even if the detected regions actually represent human heads, there is uncertainty in single camera position estimates due to sensor noise and due to assumption about the head size stated previously in this section. This necessitates evidence accumulation from multiple sensing agents to integrate their measurements to obtain a VPE.

VI. MULTI-CAMERA EVIDENCE ACCUMULATION

When a cluster leader receives a new measurement from a sensing agent, it attempts to update its set of existing position estimates by integrating the new measurement with any one of them. We now describe how this update is carried out using weighted recursive least squares technique with minimum variance.

As mentioned in the previous section, the human head position in the environment at time t is represented by the position estimate $\mathbf{p} = (\overline{\mathbf{p}}, S)$ where $\overline{\mathbf{p}}$ is the mean vector and S is the covariance matrix representing position uncertainty. Let us say that this position estimate is currently stored in a cluster leader. If a new measurement $\mathbf{p}' = (\overline{\mathbf{p}}', S')$ is received from one of the sensing agents at roughly the same time t, the cluster leader checks to see if this measurement can be integrated with the position estimate $\mathbf{p}$ by calculating the Mahalanobis distances between them:

$d_1 = \overline{\mathbf{p}}^T S^{-1} \overline{\mathbf{p}}'$ and $d_2 = (\overline{\mathbf{p}}')^T (S')^{-1} \overline{\mathbf{p}}$

If d_1 and d_2 are less than a certain distance threshold $d_{threshold}$ and if the timestamps of $\mathbf{p}$ and $\mathbf{p}'$ differ by less than a time threshold $T_{threshold}$, they are then allowed to be integrated. When $\mathbf{p}$ is updated, the new estimate is given by $\mathbf{p}_{updated} = \left(\overline{\mathbf{p}}_{updated}, S_{updated}\right)$. This calculation is carried out as follows [18]:

1) pre-computation step

$$K = S\,(S + S')^{-1} \ \ (update\,gain) \tag{4}$$

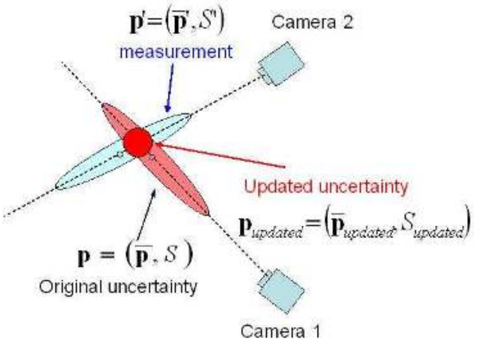

$\mathbf{p}'=(\overline{\mathbf{p}}', S')$
measurement
Camera 2
Updated uncertainty
$\mathbf{p}_{updated}=(\overline{\mathbf{p}}_{updated}, S_{updated})$
$\mathbf{p} = (\overline{\mathbf{p}}, S)$
Original uncertainty
Camera 1

Fig. 8. **Uncertainty reduction through measurement integration**

2) update step

$$\begin{aligned} \overline{\mathbf{p}}_{updated} &= \overline{\mathbf{p}} - K\,(\overline{\mathbf{p}} - \overline{\mathbf{p}}') && (5) \\ S_{updated} &= (I - K)\,S && (6) \end{aligned}$$

Since there is a time stamp associated with each position estimate, the time stamp for $\mathbf{p}_{updated}$ is calculated as the average of the time stamps for $\mathbf{p}$ and $\mathbf{p}'$. Integration of one or more measurements results in a CPE and the cluster leader keeps track of how many measurements are integrated into each CPE. In our current implementation, if three or more measurements can be integrated, a CPE becomes validated and is called VPE. Upon validation, all the intermediate CPEs that share any measurement with a VPE are eliminated. This is done primarily to ensure that each measurement only contributes to one VPE in order to minimize false detections. Additionally it leads to efficient memory usage in the cluster leader and faster integration process because there are fewer CPEs to keep track of for the purpose of dealing with a new measurement.

The evidence accumulation and position validation calculations can be understood better with the help of Figure 9. In this figure, the possible candidate position estimates gleaned from the three cameras labeled A, B and C are: $P_0(A_0, C_1)$, $P_1(A_1, B_1, C_2)$, $P_3(B_0, C_2)$, $P_4(A_1, B_1)$, $P_5(A_1, C_2)$ and $P_6(B_1, C_2)$. It is clear that the measurements A_1, B_1 and C_2 can be integrated to obtain a VPE P_1. All other

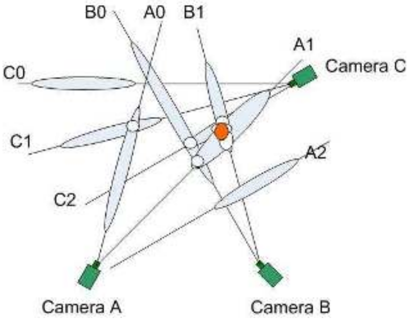


Fig. 9. **Integration and validation of position estimates.**

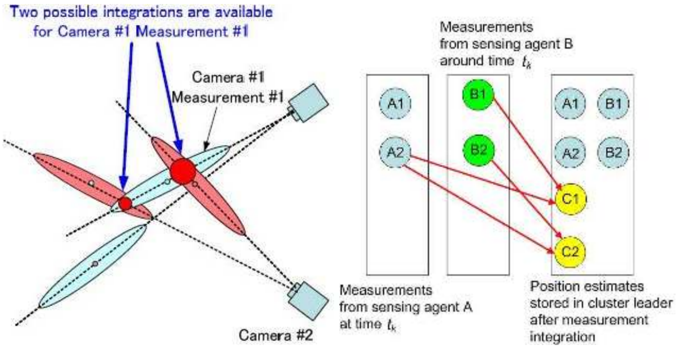

Two possible integrations are available for Camera #1 Measurement #1
Camera #1 Measurement #1
Camera #2
Measurements from sensing agent B around time t_k
A1
A2
B1
B2
A1
B1
A2
B2
C1
C2
Measurements from sensing agent A at time t_k
Position estimates stored in cluster leader after measurement integration

Fig. 10. **This figure shows the case when a single measurement from one of the cameras participates in multiple integrations vis-a-vis other camera measurements**.

CPEs containing at least one of A_1, B_1 or C_2 will then be eliminated. In the figure, the position estimates $P_2(A_1, B_0)$, $P_3(B_0, C_2)$, $P_4(A_1, B_1)$, $P_5(A_1, C_2)$ and $P_6(B_1, C_2)$ are eliminated.

The integration of measurements is based on computing the Mahalanobis distance but there may be scenarios where this leads to false detections. For example, as shown in Figure 10a, measurement #1 from camera #1 may be integrated with multiple measurements of camera #2 and all except one integration will result in false positives in head detection. In order to handle this situation, the cluster leader retains all the original measurements even after they are used to generate an updated position estimate. The measurements are retained until the measurements either become part of a VPE or are discarded as explained previously. To illustrate this, Figure 10b shows that measurement A2 from sensing agent A may be integrated individually with measurements B1 and B2 from sensing agent B, leading to updated position estimates C1 and C2 respectively. Therefore the cluster leader retains A2, B1, B2, C1 and C2 because at this point none of them is validated. The cluster leader does not know apriori which of the combinations of measurements will get validated.

A cluster leader maintains two types of data records, called the Candidate Position Estimate Box and the Validated Position Estimate Box, that are updated upon the arrival of messages from other agents. Figure 11 depicts the data records and also the data flow in a cluster leader. The data records rVBox, vBox and sVBox, all of type Validated Position Estimate Box, store information related to VPEs, and the data records rCBox, cBox and sCBox, all of type Candidate Position Estimate Box, store information related to CPEs. The rVBox and rCBox act as input buffers to receive measurements from the sensing agents or the position estimates from the lower layer cluster leaders. Similarly, the sVBox and sCBox act as output buffers to send newly validated position estimates or newly received CPEs to an upper layer cluster leader. The cBox and vBox store the current set of position estimates. A cluster leader hangs on to the "current" position estimates, that is, the estimates that are within a certain time period in the past (as a short-term memory). This is done to account for the fact that the measurements from the different sensing agents may arrive at slightly different times due to the asynchronous nature of image capture or because of communication delays in the network. To compensate for the time delay between the measurements from the different sensing agents, the integration process uses the timestamp information in addition to the Mahalanobis distances so that the integration only involves the position estimates whose time stamps are all within a certain interval. The outdated measurements or position estimates are discarded. As mentioned in section IV-B, all measurements that can not be validated even by the highest level cluster leader are also discarded.

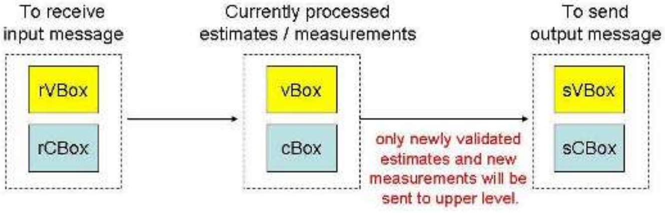
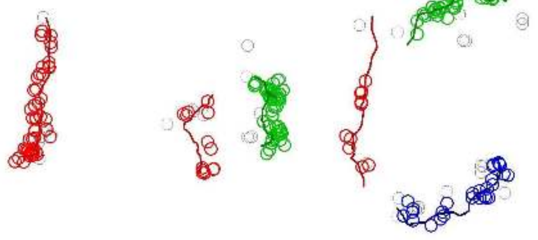

To receive input message
Currently processed estimates / measurements
To send output message
rVBox
rCBox
vBox
cBox
sVBox
sCBox
only newly validated estimates and new measurements will be sent to upper level.

Fig. 11. **The data records internal to a cluster leader.**

VII. EXPERIMENTS AND RESULTS

Our agent-based architecture was implemented using standard PCs (Pentium 4, 3.2 GHz) and 12 cameras (640x480 Dragonfly2, Point Grey Research Inc.). In order to evaluate our system for human head detection, we acquired a video sequence, approximately 2 minutes long (frame rate = 7.5 fps), of a scene in which up to three persons were moving around in a rectangular monitoring area. For analyzing the head detection performance, we considered separately the three scenarios where either only one, or just two, or all three persons were present in the monitoring area. Thirty multi-frames (time duration = 4 seconds) were extracted from the video sequence for each of these scenarios, where one multi-frame consists of 12 images, one from each of the 12 cameras, with all the images captured at approximately at the same time. Therefore in total, we used 90 multi-frames of data that corresponds to a 12 second interval. As mentioned earlier, the goal of this paper is only to *demonstrate the detection and localization performance of the system and not the tracking performance. Even in a short interval of 12 seconds, there are about 1500 candidate head regions in the ground truth data (see below) that, we believe, are adequate to demonstrate the intended performance. Therefore we can justify using short duration data for system evaluation.*

Experimental values of $d_{threshold}$ ranged from 4 to 6 and $T_{threshold} = 1/7.5$ when the frame rate is 7.5 fps. Since the Mahalanobis distance is normalized in terms of standard deviation, choosing $d_{threshold}$ between 4 and 6 seems to



(a) 1 person (b) 2 persons (c) 3 persons

Fig. 12. **The ground truth trajectories and the detected head positions reported by the system.** Black circle represent false positives

	Configuration 1			Configuration 2		
Scenario	before	after	reduction	before	after	reduction
1 person	19	6	68.4 %	19	11	42.1 %
2 person	39	14	64.1 %	39	18	53.8 %
3 person	63	19	69.8 %	63	23	63.5 %

Table I. **Comparison of the number of false positives before and after measurement integration and validation**

be high. In our experiments, we used a conservatively low estimate for the initial uncertainty in the head position (u, v, d), therefore we must set $d_{threshold}$ to a high value to ensure that the measurements corresponding to the same true head position can be integrated. The choice of $T_{threshold}$ is intuitive because we only want to integrate measurements whose temporal separation is within one frame.

For the purpose of evaluation, ground truth was generated by manually overlaying circles on human heads in single camera images. These regions were then integrated to generate ground-truthed 3D positions using weighted recursive least squares with minimum variance. Since each person was assigned a unique identity in the ground truth data, we generated motion trajectories of the individual persons by linearly interpolating between the ground truthed 3D positions. Two different configurations of sensing agents and cluster leaders were considered during the experiments: (a) Configuration 1 has a flat structure where all the 12 sensing agents are connected to single cluster leader and (b) Configuration 2 has a hierarchical structure where the 12 sensing agents are divided into four clusters of three agents each, as shown in Figure 2. Each of the four clusters have their own cluster leaders and these cluster leaders are connected to a second level cluster leader.

A numerical measure of the detection and localization performance of the system is presented in terms of 1) the number of false positives before and after measurement integration and validation (that is, evidence accumulation) in the cluster leaders; 2) calculation of the percentage of the true positives after the measurement integration and validation process; and 3) the localization accuracy of correctly detected heads. A correctly detected head in 3D is one whose shortest distance from the ground truth trajectory is less than 25 cm. The performance is estimated over 30 multi-frames for each of the 3 scenarios.

	Configuration 1		Configuration 2	
Scenario	validated heads (correctly detected heads)	% true positives	validated heads (correctly detected heads)	% true positives
1 person	56 (50)	89.3%	75 (64)	85.3%
2 person	72 (58)	80.6%	85 (67)	78.8%
3 person	88 (69)	78.4%	97 (74)	76.2%

Table II. **True positive performance after measurement integration and validation in the cluster leaders.**

Scenario	Configuration 1	Configuration 2
1 person	13 cm	15 cm
2 person	14 cm	14 cm
3 person	13 cm	13 cm

Table III. **Mean localization error in the detected head positions in the world coordinate frame.**

Figure 12 graphically illustrates the head detection results for the three scenarios described earlier. The solid curves represent the trajectories generated from the ground truth positions and the circles represent the head detections reported by the system after measurement integration and validation. The black circles denote the false positives. For the 2- and 3-person scenarios, there are some instances of missed detection. This is because of the complicated background in our test environment that results in lot of spurious contours in foreground objects. This causes the single camera head detection algorithm to perform sub-optimally, that is, with a large number of false positives.

Table I presents a comparison of the number of false positives in 2D camera images vs the 3D head positions obtained after the integration and validation of 2D measurements. For configure 1 of the agent hierarchy, the reduction in the number of false positives is approximately 64-70 % for the three scenarios. On the other hand, for configuration 2 of the agent hierarchy, the reduction in the number of false positives is roughly in the 42-64 % range. This indicates that there is a greater decrease in the false detections when a larger number of sensing agents can simultaneously participate in the measurement integration and validation process. In table II, we present the true positive detection performance. The cluster leaders integrate the measurements received from the sensing agents and generate validated position estimates (VPEs). Not all of these VPEs will be actual human head positions because sometimes false positive 2D measurements may get integrated to give a false positive VPE. But as the high true positive percentages in the table indicate, the system is very effective in filtering out false positive 2D measurements. This is so because the system needs evidence from at least three sensing agents for generating a VPE. Even if one sensing agent generates a false positive measurement, if it is not corroborated by at least two other measurements from other sensing agents, it will not go past the integration and validation stage. Table III summarizes the mean localization error of the correctly detected heads in the world coordinate frame.

We can observe from the results that both configurations have comparable detection performances. This is as per our expectations because the core evidence accumulation algorithm (equations 4, 5 and 6) does not depend on hierarchical or flat structure of the agent architecture. Nonetheless, the choice of hierarchical configuration is strongly favored by considerations of the monitoring area, scalability, real time performance, and so on. While for a small monitored area where only a few sensing agents are required, we can opt for configuration 1 due to its simple implementation. But for a large monitoring area, we would need a large number of sensing agents, which in the case of wireless sensor networks, may not be able to communicate with a single cluster leader due to limited communication range. Therefore a hierarchical structure of multiple cluster leaders becomes essential. Additionally, the measurement integration and validation process is $O(n)$ for configuration 1 and $O(log\, n)$ for configuration 2. Therefore the latter configuration is preferable for real time performance. This configuration is also more scalable because multiple sensing agents and cluster leaders can be added in a hierarchical fashion without affecting the performance of the other parts of the network.

VIII. CONCLUSIONS

In this paper, we presented a novel evidence accumulation framework for detecting and localizing humans in an indoor environment with a network of cameras. Our framework uses an agent-based architecture that can easily be scaled up as cameras are added to the network to cover a larger area. The two different types of agents we discussed in detail are the sensing agent and the cluster leader agent, the former for acquiring and locally processing the 2D images of the monitored environment and the latter for carrying out measurement integration and validation to generate accurate 3D head positions. A cluster leader agent integrates the sensing agent measurements using a weighted recursive least squares technique with minimum variance to obtain validated head position estimates. The work we reported in this paper focused on human head detection and localization in the environment. Our future research will focus on uniquely identifying different humans and tracking their trajectories in real time. An important assumption we made for the experimental results reported here was that, on the average, the human head is roughly 10 inches in diameter. This assumption can be eliminated in future improvements by using face detection in the images and using the detected faces to estimate the actual head size of the subjects. Face detection can be performed in a camera view that captures the frontal position of the subject (see, for example, [15] for detecting the orientation of a person) and then the head size information can be transmitted to the other sensing agents over the network.

IX. ACKNOWLEDGMENTS

The authors would like to thank Olympus Corporation for funding this research.

Appendix A
Position estimation in world frame

We describe the steps to obtain the CPE $\mathbf{p} = (x, y, z)$ and its error covariance matrix S from the raw measurements (u, v, d). Let $\mathbf{q} = (u, v, d)$ be the measurement vector of the human head which is a random vector with mean as the actual measurement $\hat{\mathbf{q}}$ and the error covariance matrix Q.

Let (R, T) represent the transformation for a camera from the camera coordinate frame to the world coordinate frame. Note that all the cameras are calibrated with respect to the world frame. Let $P\,(x, y, z)$ be the 3D point specified in the world frame and $P_c\,(x_c, y_c, z_c)$ be the corresponding point in the camera frame. Then

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = R \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} + T \tag{7}$$

Since the camera is calibrated, in the camera coordinate frame, we have the following relationship between the measurement (u, v, d) and the 3D point $P_c\,(x_c, y_c, z_c)$:

$$\begin{cases} u = \alpha_u \frac{x_c}{z_c} + u_0 \\ v = \alpha_v \frac{y_c}{z_c} + v_0 \\ d = \sqrt{x_c^2 + y_c^2 + z_c^2} \end{cases} \tag{8}$$

where (α_u, α_v)represent the magnification factors in the x- and y- direction, and (u_0, v_0) represent the image center of the camera image. From (8), we obtain

$$\begin{cases} z_c = \frac{d}{\sqrt{\left(\frac{u-u_0}{\alpha_u}\right)^2 + \left(\frac{v-v_0}{\alpha_v}\right)^2 + 1}} \\ x_c = \frac{u-u_0}{\alpha_u} z_c \\ y_c = \frac{v-v_0}{\alpha_v} z_c \end{cases} \tag{9}$$

Using Eqs. (7) and (9), we can compute (x, y, z).

Appendix B
Error Covariance estimation in World Frame

We will now show how the relationship between the measurement vector $\mathbf{q} = (u, v, d)$ and its world-coordinate version $\mathbf{p} = (x, y, z)$ can be used to transform the error covariance matrix Q associated with $\mathbf{q}$ into the error covariance matrix S associated with $\mathbf{p}$. Let $\mathbf{p} = \mathbf{f}\,(\mathbf{q})$ represent the transformation from $\mathbf{q}$ to $\mathbf{p}$ which is actually a non-linear transformation. Consider first the mean vector of $\mathbf{p}$:

$$\overline{\mathbf{p}} = E\,[\mathbf{p}] = E\,[\mathbf{f}\,(\mathbf{q})] \approx \mathbf{f}\,(E[\mathbf{q}]) = \mathbf{f}\,(\overline{\mathbf{q}}) = \mathbf{f}\,(\hat{\mathbf{q}}) \tag{10}$$

where we have used the linear approximation in writing $E\,[\mathbf{f}\,(\mathbf{q})] \approx \mathbf{f}\,(E[\mathbf{q}])$. This approximation linearizes the nonlinear function by retaining only the first term in the Taylor series expansion. The nonlinear function $\mathbf{f}(.)$ can be expanded as a Taylor series about $\hat{\mathbf{q}}$:

$$\mathbf{p} = \mathbf{f}(\mathbf{q}) = \mathbf{f}(\hat{\mathbf{q}} + \delta\mathbf{q}) = \mathbf{f}(\hat{\mathbf{q}}) + higher\ order\ terms$$
$$\Rightarrow \mathbf{p} \approx \mathbf{f}(\hat{\mathbf{q}})$$
$$\Rightarrow E\left[\mathbf{f}(\mathbf{q})\right] = E\left[\mathbf{p}\right] = E\left[\mathbf{f}(\hat{\mathbf{q}})\right] = \mathbf{f}(\hat{\mathbf{q}}) = \mathbf{f}\left(E[\mathbf{q}]\right)$$

As for the covariance matrix S, we consider the deviation from the mean vector. For $\delta\mathbf{p} = \mathbf{p} - \bar{\mathbf{p}}$, $\delta\mathbf{q} = \mathbf{q} - \bar{\mathbf{q}} = \mathbf{q} - \hat{\mathbf{q}}$, we obtain (again using the linear approximation)

$$\delta\mathbf{p} = \frac{\partial \mathbf{f}}{\partial \mathbf{q}} \delta\mathbf{q} \tag{11}$$

The covariance matrix S is now computed as follows:

$$\begin{aligned}
S &= E\left[\left(\mathbf{p} - \overline{\mathbf{p}}\right)\left(\mathbf{p} - \overline{\mathbf{p}}\right)^T\right] \\
&= E\left[\delta\mathbf{p}\delta\mathbf{p}^T\right] \\
&= E\left[\frac{\partial \mathbf{f}}{\partial \mathbf{q}} \delta\mathbf{q}\delta\mathbf{q}^T \left(\frac{\partial \mathbf{f}}{\partial \mathbf{q}}\right)^T\right] \\
&= \frac{\partial \mathbf{f}}{\partial \mathbf{q}} E\left[\delta\mathbf{q}\delta\mathbf{q}^T\right] \left(\frac{\partial \mathbf{f}}{\partial \mathbf{q}}\right)^T \\
&= \frac{\partial \mathbf{f}}{\partial \mathbf{q}} E\left[\left(\mathbf{q} - \overline{\mathbf{q}}\right)\left(\mathbf{q} - \overline{\mathbf{q}}\right)^T\right] \left(\frac{\partial \mathbf{f}}{\partial \mathbf{q}}\right)^T \\
&= \frac{\partial \mathbf{f}}{\partial \mathbf{q}} Q \left(\frac{\partial \mathbf{f}}{\partial \mathbf{q}}\right)^T
\end{aligned}$$

where $\frac{\partial \mathbf{f}}{\partial \mathbf{q}}$is evaluated at $\mathbf{p} = \overline{\mathbf{p}}, \mathbf{q} = \overline{\mathbf{q}} = \hat{\mathbf{q}}$.

C. REFERENCES

[1] J. Black and T. Ellis, "Multi camera image tracking," *Image and Vision Computing*, , no. 24, pp. 1256–1267, 2006.

[2] A. Nakazawa, H. Kato, and S. Inokuchi, "Human tracking using distributed vision systems," *Pattern Recognition, 1998. Proceedings. Fourteenth International Conference on*, vol. 1, pp. 593–596 vol.1, 1998.

[3] Q. Cai and J.K. Aggarwal, "Tracking human motion in structured environments using a distributed-camera system," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 21, no. 11, pp. 1241–1247, 1999.

[4] S.L. Dockstader and A.M. Tekalp, "Multiple camera tracking of interacting and occluded human motion," *Proceedings of the IEEE*, vol. 89, no. 10, pp. 1441–1455, Oct 2001.

[5] J. Krumm, S. Harris, B. Meyers, B. Brumitt, M. Hale, and S. Shafer, "Multi-camera multi-person tracking for easyliving," *Visual Surveillance, 2000. Proceedings. Third IEEE International Workshop on*, pp. 3–10, 2000.

[6] Jinman Kang, I. Cohen, and G. Medioni, "Continuous tracking within and across camera streams," *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, vol. 1, pp. I–267–I–272 vol.1, 2003.

[7] S. Khan and M. Shah, "Consistent labeling of tracked objects in multiple cameras with overlapping fields of view," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 25, no. 10, pp. 1355–1360, 2003.

[8] O. Javed, Z. Rasheed, O. Alatas, and M. Shah, "Knight/spl trade/: a real time surveillance system for multiple and non-overlapping cameras," *Multimedia and Expo, 2003. ICME '03. Proceedings. 2003 International Conference on*, vol. 1, pp. I–649–52 vol.1, 2003.

[9] V. Kettnaker and R. Zabih, "Bayesian multi-camera surveillance," *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, vol. 2, pp. –259 Vol. 2, 1999.

[10] R.T. Collins, O. Amidi, and T. Kanade, "An active camera system for acquiring multi-view video," *Image Processing. 2002. Proceedings. 2002 International Conference on*, vol. 1, pp. I–527–I–520 vol.1, 2002.

[11] T. Chang and S. Gong, "Tracking Multiple People with a Multi-Camera System," *IEEE Workshop on Multi-Object Tracking (WOMOT'01)*, p. 0019, 2001.

[12] F. Porikli and A. Divakaran, "Multi-camera calibration, object tracking and query generation," *Multimedia and Expo, 2003. ICME '03. Proceedings. 2003 International Conference on*, vol. 1, pp. I–653–6 vol.1, 2003.

[13] R. Collins, A. Lipton, H. Fujiyoshi, and T. Kanade, "Algorithms for cooperative multisensor surveillance," vol. 89, no. 10, pp. 1456–1477, October 2001.

[14] A. Mittal and L. S. Davis, "M2Tracker: A Multi-View Approach to Segmenting and Tracking People in a Cluttered Scene," *International Journal of Computer Vision*, vol. 51, no. 3, pp. 189–203, 2003.

[15] A. Utsumi, H. Mori, J. Ohya, and M. Yachida, "Multiple-human tracking using multiple cameras," *Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on*, pp. 498–503, 1998.

[16] A. Utsumi, H. Yang, and J. Ohya, "Adaptive human motion tracking using non-synchronous multiple viewpoint observations," *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, vol. 4, pp. 607–610 vol.4, 2000.

[17] J. Park, A. Tabb, and A. C. Kak, "Hierarchical Data Structure for Real-Time Background Subtraction," *Proceedings of IEEE International Conference on Image Processing*, 2006.

[18] Y. Bar-Shalom and Xiao-Rong Li, *Estimation and Tracking: Principles, Techniques and Software*, Artech House, Inc., 1993.

[19] L. Zhao, *Dressed Human Modeling, Detection, and Parts Localization*, Ph.D. thesis, Carnegie Mellon University, Pittsburgh, PA, 2001.