Identifying People in Camera Networks using Wearable Accelerometers

Thiago Teixeira, Deokwoo Jung, Gershon Dublon and Andreas Savvides
Yale University
10 Hillhouse Ave
New Haven, CT
thiago.teixeira@yale.edu

ABSTRACT
We propose a system that identifies people in a sensor network. The system fuses motion information measured from wearable accelerometer nodes with motion traces of each person detected by a camera node. This allows people to be uniquely identified with the IDs the accelerometer-node that they wear, while their positions are measured using the cameras. The system runs online in real time, with high precision and recall results. A prototype implementation using iMote2s with camera boards and wearable TI EZ430 nodes with accelerometer sensorboards is described and evaluated.

Categories and Subject Descriptors
C.3 [Special-Purpose and Application-Based Systems]: Real-time and embedded systems; I.4 [Image Processing and Computer Vision]: Scene Analysis—Sensor fusion, Tracking

1. INTRODUCTION
A large obstacle to the deployment of assisted-living systems in multiple-person or family homes is the problem of differentiating between people and identifying them to properly attend their individual needs. For this reason, much of the current assisted-living technology focuses on single-person scenarios — and often break as soon as visitors are invited into the home. Additionally, multiple-person homes present complex privacy requirements for assistive technologies: the only people whose information is captured and stored should be the ones who choose to utilize the system. Meanwhile, cameras are becoming increasingly popular sensors in assistive environments, given their long sensing range and ability to measure distinct information modalities (such as location, pose, motion path and ambient lighting). However, the problem of associating detected people across multiple image frames as well as robustly identifying them solely with visual features is still a topic of much research in computer vision.

2. RELATED WORK
Accelerometers and cameras are often combined to track the camera motion, generally for use in robot navigation [1], and virtual or augmented reality [2]. In such cases, the accelerometer is placed on the same rigid object as the camera, which moves in relation to its environment. This contrasts with the setup described in this paper, where the camera is stationary and accelerometers are placed on the moving people in the camera’s field-of-view (FOV) in order to identify...
accelerometer

that maximizes a similarity measure. Therefore, if the matching between accelerometers and detected locations is the core of the identification problem can be described as finding the detected locations of people with a camera network to associate problem as input. For this purpose, we make use of the fact that correct tracks must belong the multi-dimensional association problem, and it is known to be NP-hard [9]. One of the seminal works in this area is the multiple-hypothesis tracking algorithm [10]. When only the position of each detected person is used to perform this association, this is called the motion correspondence problem and is the subject of much research [11]. Other times, additional image features (such as size, color, shape or motion gradient) [12][13] or motion models are used to offer additional clues regarding frame-to-frame associations, but usually with mixed results in uncontrolled environments. In contrast, the algorithm presented here is lightweight, does not make assumptions about motion models, and does not require an exact solution of the motion correspondence problem and is the subject of much research.

The problem we solve in this paper is the matching of detected locations of people with a camera network to their accelerometer signals, to obtain location-ID pairs. The core of the identification problem can be described as finding the matching between accelerometers and detected locations that maximizes a similarity measure. Therefore, if $Z_k$ is the set of all accelerometer measurements at time $k$, and $X_k$ is the set of all detected locations, then at each time $k$ we must find the match matrix $M_k$ through the equation below:

$$
\arg\max_{M_k} \sum_{i=1}^{\left|Z_k\right|} \sum_{j=1}^{\left|X_k\right|} f(z_{ik}, x_{jk})M_k^{ij}
$$

where $z_{ik}$ is the $i^{th}$ accelerometer measurement at time $k$, and $x_{jk}$ is the $j^{th}$ detected position at that time. Note that the index $i$ of the accelerometer measurements is the ID of the nodes that transmitted them, while the $j$'s are random internal IDs of each detected person without any physical relevance. The match matrix $M_k$ is a matrix of size $|Z_k| \times |X_k|$ describing the associations between accelerometers and detected people in the image frame. Since the same person cannot be wearing two accelerometers, and the same accelerometer cannot be in more than one place at a time, $M_k$ must follow a few constraints:

$$
M_k^{ij} = \begin{cases} 
1 & \Rightarrow z_{ik}, x_{jk} are associated \\
0 & \Rightarrow no association 
\end{cases}
$$

$$
M_k^{ij} = \begin{cases} 
M_k^{ij} = 0 \forall \ell \in [1,|Z_k|], \ell \neq j \\
M_k^{ij} = 0 \forall \ell \in [1,|X_k|], \ell \neq i 
\end{cases}
$$

Despite the brief definition, the problem that is targeted in this paper cannot be solved by directly associating accelerometers to detected people as described in Equation 1. Instead, the two types of measurements (accelerations and positions) must be brought to a common representation in order to be compared, which leads to an exponentially complex problem. As shown in Figure 2, our solution is divided into two parts:

**Identification** — In order to match the motion data from the wearable accelerometers with detections from the camera nodes, we transform each into a signal that is proportional to the person’s floor-plane acceleration. We, then, measure their similarity by computing their correlation coefficient. This is described in Section 3.1. To obtain these measurement measurements from the detected locations, however, we must first obtain a time-series of locations for each person in the scene (tracks). This is called the multi-dimensional association problem, and it is known to be NP-hard [9].

**Association** — Rather than solving the association problem, we make use of the fact that correct tracks must belong to real people in the scene, and therefore must correlate with some accelerometer (exactly one, in fact). We use this to approximate the multi-dimensional association problem as...
3.1 Base-Case for Identification Problem: 1 Person in FOV, 2 Accelerometers

Given that the communication range of a sensor node can be larger than the camera’s FOV (depending on the camera orientation and occlusions), it is possible for multiple accelerometers to be detected while only a single person is present in the FOV. In this case, there are no frame-to-frame association ambiguities: if it is known a priori that there is exactly one person in the scene, then it is known that the person detected in image frame \( I_k \) at time \( k \) is the best match, so \( ID = 2 \) should be assigned to the detected person.

a one-dimensional association with polynomial complexity. This is described in Section 3.2.

Figure 3: Base case described in Section 3.1, where a single person is in the camera’s field of view, while two accelerometers are within communication range. In this example, the second measured acceleration is the best match, so \( ID = 2 \) should be assigned to the detected person.

Figure 4: Superimposition of aligned signals from accelerometer and camera, showing the approximate proportionality between them.

format. This process is known as data alignment [14]. We align the two signals to the same temporal frame of reference by associating all position measurements that belong to the same person into a time series. From this time series, the person’s acceleration in the image plane can be easily extracted by double differentiation, as shown in Figure 3. We also align the accelerometer measurements into a signal proportional to the overall body acceleration by calculating the magnitude of the 3D acceleration vector and finding the envelope of the signal to remove noise caused by the stepping motion and by accelerometer-bouncing artifacts [15][16]. Figure 4 shows the similarity between two matching signals that were processed in this manner. These two signals are proportional to the person’s floor-plane acceleration, and, therefore, also proportional to one another.

If \( \alpha \) and \( \beta \) are the functions which align accelerometer and camera signals into the same common representation, then the similarity between the two signals can be calculated by detecting whether the two signals are proportional using Pearson’s correlation coefficient \( r \):

\[
g(\alpha(z^k_\ell), \beta(x^k_\ell)) = r(\alpha(z^k_\ell), \beta(x^k_\ell))
\]

where \( \alpha(z^k_\ell) \) is a track containing a time series of consecutive person detections \( \alpha(z^k_\ell) = (z^{k_0}_\ell, ..., z^{k_n}_\ell) \) with \( n \in \mathbb{N} \) and \( 0 < n < k \). Figure 5(a) shows the experimental value of the correlation coefficient between 5 tracks and 2 accelerometer signals. The correct matches can be easily seen given their large correlation coefficients.

Note, however, that \( g : Z_k \times \Theta_k \rightarrow \mathbb{R} \) from Equation 4 has a different domain than the similarity function \( f \) from Equation 1. To assign IDs to detected people using \( g \), the maximization problem in Equation 1 must be modified to use tracks rather than person detections:

\[
\arg\max_{\Omega_k} \sum_{i=1}^{z_k} |\Theta_k| \sum_{\ell=1}^{\theta_k} g(\alpha(z^i_\ell), \beta(x^i_\ell))\Omega_k^{i\ell}
\]

where \( \Theta_k = \{\theta_k^i\} \) is the set of all tracks at instant \( k \), and \( \Omega_k \) is a match matrix associating accelerometer signals to tracks. The matrix \( \Omega_k \) follows similar rules as \( M_k \) (Equation 3) but it additionally does not allow the same detected person to be assigned to multiple tracks at any time instant. So \( \Omega_k \) must follow the additional rule that for any two elements equal to 1, the corresponding tracks must have an empty intersection:

\[
\Omega_k^{i\ell_1} = \Omega_k^{i\ell_2} = 1, \ell_1 \neq \ell_2 \implies \theta_k^{i\ell_1} \cap \theta_k^{i\ell_2} = \emptyset
\]
We call this the “strong no-intersection” property, which will be relaxed in Section 4 in order to approximate the solution for real time operation.

### 3.2 Base-Case for Association Problem: 2 People in FOV with Accelerometers

For scenarios where there is more than one person in the FOV, or where there are false-positive detections, generating correct tracks from a sequence of person detections can be quite complex, and there may be no unique solution. Moreover, the number of tracks in a multiple-person case quickly becomes unmanageable. To solve the problem in Equation 5 within realistic time and memory constraints we use the similarity function $g$ in Equation 4 not only to match accelerometer to tracks but also to guide the track creation process. The aim is to keep a small set of realistic track hypotheses by using $g$ rather than keeping a comprehensive set of all possible tracks.

Consider a simple scenario where it is known that exactly two people are in the FOV, and they are within a large distance of one another. In this case, it is easy to infer their position histories by creating tracks connecting each detection at frame $k$ to the only detection at $k + 1$ that is within a physically plausible displacement threshold — a process known as gating. However, tracking ambiguities arise when the two people are close to one another (Figure 6), and multiple competing track hypotheses are possible. If all possible track hypotheses are considered by the tracking algorithm, then due to combinatorial explosion the complexity of the problem quickly becomes unmanageable. This is shown in Figure 7, where two people meet for 6 time instants (at 15 frames per second this corresponds to 0.4s) generating more than 64 hypotheses.

Thus, only tracks within the gate are considered (Figure 6) generating more than 64 hypotheses. It is known that there are exactly $N$ people in the FOV, then the number of hypotheses after $K$ ambiguous frames is $N!^K$. If the people are allowed to enter/leave, and a realistic detector is assumed (with the possibility of false positive detections), then the number is even larger. This association problem can be described as selecting the set of tracks that, at each time instant $k$, globally minimizes some distance metric $h$:

$$
\arg \min_{\Theta_k} \sum_{\ell=1}^{K} \sum_{j=1}^{|X_k|} h(\theta_{k-1}^{l}, x_{k}^{j}) \Phi_{k}^{ij}
$$

(7)

where $\Phi_{k}$ is a match matrix that follows the same rules as $M$ in Equation 3 (which causes the constructed tracks to naturally follow the strong no-intersection rule from Equation 6). From $\Phi_{k}$, the set of tracks $\Theta_{k}$ which solves Equation 7 for time instant $k$ is directly obtained. The simplest similarity metric for track-to-location association is the Euclidean distance between the track’s latest location $x_{k}^{n}$ and the detection $x_{k}^{L}$:

$$
h(\theta_{k-1}^{l}, x_{k}^{j}) = \text{dist}(x_{k-1}^{n}, x_{k}^{j})
$$

(8)

where $n$ is the track length. This is often called nearest-neighbor association.

As described before, it is not tractable to exactly solve Equation 7 due to the expansive number of tracks. Luckily, to identify the people in the scene it is not necessary to solve this complex association problem, since the no-intersection property is handled later in the process by the maximization in Equation 5. So we bypass this problem by generating several conflicting track hypotheses ($\Theta_{k}$), rather than finding the best non-conflicting solution ($\Theta_{k}$). The set $\Theta_{k}$ of track hypotheses is defined to contain all tracks that pass a goodness criterion:

$$
\Theta_{k} = \left\{ \theta_{k}^{l} \in \Theta_{k-1} \times X_{k} : h(\theta_{k-1}^{l}, x_{k}^{j}) < \tau_{a}, \exists z_{k}^{l} \in Z_{k} | g(z_{k}^{l}, \theta_{k}^{l}) > \tau_{r} \right\}
$$

(9)

where $\tau_{a}$ and $\tau_{r}$ are thresholds that filter out bad hypotheses. Thus, only tracks within the gate are considered ($h(\cdot, \cdot) < \tau_{a}$). Moreover, the similarity measure $g$ is used in a manner analogous to the use of additional image attributes (size, color, shape) and motion models that are usually employed in multiple-target trackers. In this case, we keep only the tracks that can be explained to some degree by at least one of the accelerometer signals. This is described in greater detail in Section 4.2. Of course, image and motion attributes from the literature can be used in addition to the accelerometer signal, for increased robustness if necessary.

### 4. PERSON-IDENTIFICATION ALGORITHM

From the observation that the person-identification problem is naturally composed of two interconnected parts (association and identification), we design our algorithm as a cycle consisting of two blocks: a tracker and a comparator. The tracker generates a set $\Theta_{k}$ of tracks from sequences of person detections, filtering them according to the pa-
The comparator then passes the set $\Theta^k_\tau$ in Equation 5, assigning IDs to each detected people. The comparator is in charge of one-dimensional association problem from Equation 5 with the algorithm, and is shown is Figure 8.

Although in the problem description we discussed the similarity between each track and each accelerometer, $g(z^k_i, \theta^k_\tau)$, this is not exactly how things take place in the identification algorithm. Instead, each track is marked as belonging to a specific accelerometer, which is the only one it will ever be compared to. The reason for this is that the correlation coefficient requires the two input signals to be of the same length. When each track is created, we must bootstrap the sufficient statistics to compute its correlation with each specific accelerometer. This also allows us to keep the complexity low, since tracks that have historically not correlated well with a given accelerometer can be pruned and never compared to that accelerometer again.

Other than this, the algorithm takes two main approaches to allow real time operation: (1) it simplifies the accelerometer-to-track assignment problem in Equation 5 by weakening the track intersection property (Section 4.1); (2) it restrains the number of track hypotheses to a minimum through several means, as described in Section 4.2.

### 4.1 Relaxing the No-Intersection Property

Since our algorithm aims to provide the best immediate results without the intention to reconstruct past traces, we relax the strong no-intersection constraint of Equation 6 to require only that the \textit{newest} position measurements in each matched track to not intersect. That is, the following weak no-intersection constraint is used instead:

$$\Omega^k_\tau = \Omega^k_{\tau_2} = \emptyset$$

Although this relaxes the strong no-intersection property of Equation 6, the similarity measure $g$ used in the identification (Equation 5) guarantees that tracks correlate well with their matched accelerometer. So, as long as the motion of the people in the scene is not correlated with one another, most tracks selected by Equation 10 will still be strongly non-intersecting. In the case that their motion is correlated, then it is not possible to identify them based on motion characteristics alone, whether strong non-intersection is enforced or not. Hence, this simplification has little negative effect on the quality of the tracks, while greatly limiting the problem’s complexity.

### 4.2 Controlling the Number of Hypotheses

**Combinatorial Contention** — When there are ambiguous situations, such as in Figure 7, the number of tracks grows exponentially. In order to contain this growth, we only resolve ambiguities after the people move apart. For this, the algorithm keeps track of the number of people inside each track’s gate (a circle of radius $R$). If the number is greater than one, then the track is marked as being ambiguous. Otherwise, it is marked as unambiguous. Each ambiguous track $\theta^k_{\tau_2}, \ldots, \theta^k_{\tau_{K^k}}$ gets extended into time $k$ as $\theta^k_\tau$ by assigning it the closest detection $\mathbf{x}_k$ rather than forking into one track for each within-gate detection. When a track transitions from ambiguous to non-ambiguous, however, it is forked for each detection inside a gate with radius $2R$. If $N_{\text{old}}$ is the number of people in the $2R$ gate, instead, then of ending up with $N_{\text{old}}/2$ tracks as before, each track splits into just $N_{\text{old}}/2$ alternatives, most of which are pruned within a few seconds by a track-pruning process.

**Pruning Tracks and Allowing “Leaving”** — If a track correlates badly with all accelerometer signals, then it cannot belong to an accelerometer-wearing person, and should be pruned. Figure 5(b) shows a histogram of the correlation values of correct and incorrect accelerometer-to-track assignments. It is clear from the plot that the two can be easily distinguished, and that a threshold value $\tau_r \approx 0.55$ can be used for this purpose. However, as shown in Figure 9(a), the correlation $r$ between an accelerometer and a track takes a few seconds to converge. Oftentimes the correct accelerometer-track association has a poor correlation (< $\tau_r$) for the first few seconds, which can cause correct tracks to be prematurely pruned.

For this reason, we compute the estimated correlation error as a function of track age by using confidence intervals. But since Pearson’s correlation coefficient does not have a Gaussian sampling distribution, we must first convert it with Fisher’s $z'$ transformation, for which confidence intervals can be calculated:

$$z'(r) = \frac{1}{2} \ln[(1 + r)/(1 - r)] \quad (11)$$

The standard error of $z'$ is known to be $SE = 1/\sqrt{n-3}$, where $n$ is the number of samples used in the computation of the correlation. With this, we compute the 90% confidence interval of $f$ as ranging from $z_{\text{low}}$ to $z_{\text{high}}$:

$$z_{\text{low}}(r) = r - \frac{1.645}{\sqrt{n-3}} \quad z_{\text{high}}(r) = r + \frac{1.645}{\sqrt{n-3}} \quad (12)$$

where the number 1.645 comes from the 90% confidence interval of a Normally distributed random variable (i.e. 90% of the density is within 1.645 standard deviations from the mean). Equation 9 is, then, modified to apply the $\tau_r$ threshold on $z_{\text{high}}$ instead. That is, $\alpha(.) > \tau_r$ becomes $z_{\text{high}}(.) > z'(\tau_r)$. This way, the only tracks that get pruned are those where there is a 95% confidence that the track does not correlate above $\tau_r$ (95% because the threshold acts on a single-sided confidence interval). For comparison, Figure 9(b) shows...
the \( z' \) and confidence intervals for the signals from Figure 9(a).

Since correlations of longer signals have a smaller standard error, they are inherently more trustworthy. We, therefore, prioritize longer tracks by using \( z_{\text{new}}' \) instead of \( r \) in Equation 4. So if two tracks have the same \( r \) (and, hence, the same \( z' \)) the older track will be given a higher weight in \( g \) since the \( z_{\text{new}}' \) will be higher for the older track. With this change, Equation 4 becomes:

\[
g(z_k', \theta_k') = z'(\alpha(z_k'), \beta(\theta_k')) - 1.645/\sqrt{n-3}
\]  

(13)

Note that, as the standard error cannot be computed for tracks smaller than 4 samples, we only allow a track to be pruned if its size is greater than 4. Given that new tracks are created at the end of each ambiguous period, this causes the number of tracks to depend on the number of ambiguities.

**Faster Error Recovery and Allowing “Entering”** — When a new person enters the camera FOV, a new track must be created for comparison with each accelerometer. Similarly, when a new accelerometer is detected, it must be included for comparison with each existing track. For this reason, the algorithm always keeps at least one track for each accelerometer-location combination. If one does not exist, it is created. This can happen either because a new person or accelerometer has been detected (“entering”) or because an existing track has been pruned. The end result is that tracks that may or may not represent a correct ground-truth trace are constantly created (and constantly pruned, if they do not pass the \( \tau \) threshold). This ensures that there is always one alternative for each accelerometer-location assignment, which allows for quick recovery in case a correct track becomes associated with the wrong detection due to tracking errors. This puts a lower bound of \(|Z_k| \times |X_k|\) on the number \(|\Theta_k|\) of track hypotheses at any time \( k \). When there are no ambiguities, the track-pruning process ensures that the lower-bound is reached. Hence, for most real-world cases, it is expected for the average number of track hypotheses to be close to \(|Z_k| \times |X_k|\).

5. EVALUATION

We first performed a set of offline experiments where data was gathered with a USB camera and Bluetooth inertial measurement units. These were used to verify the correctness of the algorithm independently from implementation-dependent effects, such as the performance of the person detector or of the network layer. Then, online real time experiments were performed using the iMote2 sensor node with our custom camera board [18], as well as TI EZ430-RF2480 nodes equipped with a SparkFun IMU 5DOF board, containing an Analog Devices ADXL330 accelerometer (Figure 1). The purpose of the online experiments is to demonstrate the viability of the algorithm for real time computation in actual deployments. For all of these experiments, the cameras were mounted on a 3m high ceiling, facing down. This gives a total area of \( 3m \times 2m \) where people are entirely contained in the FOV. This is the area within which the people were asked to stay. The accelerometer nodes were placed on the front of each person’s belt. The orientation of the accelerometer is unimportant, given that it is the magnitude of the 3D acceleration vector that is used in the similarity metric.

5.1 Offline experiments

Before an online evaluation of the entire system, we performed experiments to judge the performance of the algorithm by itself, without multiple-person-detection errors such as merging two people together when they are close, or splitting a single person into when part of them is too similar to the background. For this we captured five separate videos and the corresponding accelerometer traces of a single person walking in a room for around 1 minute. The person detector used in this experiment computed the person in the scene by comparing each frame to an image of the empty room (background subtraction). Since the traces were captured separately in a static, controlled environment, we were able to obtain high precision image-plane coordinates for each person by calculating the center of mass (centroid) of the foreground pixels. The accelerometers were sampled at 100Hz, and the camera at 15Hz. Time was roughly synchronized by hand, by visually matching the features from an acceleration magnitude plot for each accelerometer to a plot of the corresponding centroid’s speed.

We ran the algorithm for all different 2-person, 3-person,

<table>
<thead>
<tr>
<th>count</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>precision</td>
<td>0.951</td>
<td>0.875</td>
<td>0.790</td>
<td>0.627</td>
</tr>
<tr>
<td>recall</td>
<td>0.956</td>
<td>0.931</td>
<td>0.887</td>
<td>0.821</td>
</tr>
<tr>
<td>proc. time (s)</td>
<td>2.57</td>
<td>6.29</td>
<td>11.02</td>
<td>16.74</td>
</tr>
<tr>
<td>ambig./pers.</td>
<td>38.25</td>
<td>68.57</td>
<td>93.90</td>
<td>116.6</td>
</tr>
<tr>
<td>avg. tracks</td>
<td>3.92</td>
<td>8.81</td>
<td>15.63</td>
<td>24.39</td>
</tr>
<tr>
<td>max. tracks</td>
<td>8</td>
<td>15</td>
<td>24</td>
<td>30</td>
</tr>
</tbody>
</table>

Table 1: Experimental results for offline algorithm when 2, 3, 4 or 5 people are in the FOV at the same time.
4-person and 5-person combinations of the five traces. The centroid traces were overlaid onto the same image plane and the centroids’ internal index were randomly shuffled for each frame. We additionally simulated people entering and leaving the field of view at random times while still being in range for the accelerometer sensing. This was done by randomly cropping the beginning and end of the centroid traces, and leaving the accelerometer traces intact. For all of these, the ground truth frame-by-frame associations and absolute person IDs were known, given that the traces were acquired separately. Using the ground truth data, we calculated the following metrics: The precision is a metric answers the question: when the system identifies a person, how often is the ID assignment correct? The precision is calculated as $\frac{TP}{(TP + FP)}$, where $TP$ is the number of true positives (correct assignments) and $FP$ is the number of false positives (incorrect assignments). Recall answers the question: when a given person is in the scene, how often does the system correctly identify him? The recall is calculated as $\frac{TP}{(TP + FN)}$, where $FN$ is the number of false negatives, that is, the number of times the person was deemed absent when they were actually present.

The averaged experimental values of these two metrics are shown in Table 5. The algorithm shows strong performance for 2, 3 and 4 people. For 5 people, the precision falls under 0.7, but the recall stays high throughout. The measured processing time is, as expected, proportional to the average number of tracks. Since the number of tracks stays within the predicted value of $Z_t \times Z_b$, the processing time remains nearly constant. The number of ambiguous frames per person (ambigs/per) is also reported. Were the tree-pruning process not present, the expected number of tracks would be in the order of $X_k^{ambigs/per}$. This is at least $2 \times 10^4$ for the 2-person case, and much more for the others. The correctness of the algorithm has a stronger dependence on the ambig/person rate than on the number of people in the FOV per se.

Figure 10 shows the output of the algorithm for an example trace where 4 people were present. The track assignment plots show the ground-truth ID of the centroid that was associated by the algorithm to each accelerometer. For an accelerometer with $ID = A$ (where $A$ is some number), it is desirable for the plot to be a constant line at $y = A$. This is often the case after time enough has passed for the correlations to converge, as seen in the plots. Meanwhile, the ambiguity resolution plot (shown only for person 1, due to space restrictions in this paper) shows how often ambiguities occur (usually consisting of multiple frames at a time), and whether the algorithm is able to resolve them correctly or incorrectly. On average, ambiguities were correctly resolved 80.72% of the times. For the remaining 19.28% when the ambiguity resolution failed, the algorithm eventually found the correct assignment through the correlation metric. That is, the algorithm is able to automatically recover from incorrect hypotheses. Finally, the third type of plot in the figure, the hypotheses correlation plot, shows the $z'$ metric of the selected hypothesis (thick blue line) compared to that of the losing hypotheses for the same accelerometer (light blue). The hypotheses for other accelerometers are shown in light gray. Note how after ambiguous periods small tracks fork from the correct one. They are quickly pruned by the combinatorial contention process described in Section 4.2. You can find videos of these experiments at http://enaweb.eng.yale.edu/drupal/InertialIdentification.

5.2 Online experiments

To assess the viability of the system in an assisted-living deployment, we tested a prototype implementation consisting of an iMote2 camera node mounted on the ceiling, and two people carrying wearable EZ430 sensor nodes with accelerometers. The centroid of each foreground blob was extracted by segmenting them through 8-neighbor connected component analysis. As expected, this often resulted in the typical blob-merging and splitting artifacts that are a product of small occlusions and visual similarity with the background scene. Detections were collected into packets containing pairs of centroids and timestamps, and transmitted wirelessly to a base node. The whole process took place at a rate of around 15Hz in the sensor node, fluctuating based on the number of people in the FOV. The wearable nodes used in the experiment were programmed to sample the accelerometer at a rate of 50Hz, calculating the signal envelope locally, and transmitting it to the base through its ZigBee radio. The algorithm ran on a computer, which displayed the results live on its screen. The prototype system shows that the algorithm is able to operate in real time, under real-world conditions containing a non-ideal human detector (with frequent false positives and false negatives) and with data being transmitted live from sensor nodes.
6. CONCLUSION

We have presented an algorithm that uses infrastructure camera nodes and wearable accelerometers to identify people in a sensor network. The algorithm is demonstrated with a prototype system, achieving good precision and recall. There is no limit on the number of tag-wearing people or the number of people in the FOV. Although the approximations for real-time execution increase the number incorrect matches immediately following ambiguous periods, the algorithm is able to quickly recover.

Possible improvements include utilizing additional image features for increased robustness against ambiguities. By coupling this system with color histograms, for example, better detection rates should be easily achieved. Future work includes expanding the algorithm to make use of multiple cameras as a single seamless sensor, as well as considering deployments where there are large gaps in camera coverage. Before the system can be used in a long-term deployment, power consumption and network utilization must be properly analyzed. To this end, it is possible that an adaptive sampling and transmission scheme can be devised, which preprocesses accelerometer samples and only transmits them if it is deemed that they can significantly impact the correlation metric.

7. REFERENCES


