Toward Cooperative Localization of Wearable Sensors using Accelerometer and Camera

Deokwoo Jung
Department of Electrical Engineering
Yale University
New Haven, Connecticut 06511–0250
Email: deokwoo.jung@yale.edu

Thiago Teixeira
Department of Electrical Engineering
Yale University
New Haven, Connecticut 06511–0250
Email: thiago.teixeira@yale.edu

Andreas Savvides
Department of Electrical Engineering
Yale University
New Haven, Connecticut 06511–0250
Email: andreas.savvides@yale.edu

***Abstract*—This work describes a new approach for localizing people by cooperative sensor fusion of lightweight camera and wearable accelerometer measurements. We present the algorithm to identify people moving around as they are detected by cameras deployed in the infrastructure. The algorithm uses an appropriate correlation metric that is then used to develop an ID matching algorithm that can associate people in the scene to their global ID emitted from a wireless accelerometer sensor node worn on their belts. First we conduct a set of preliminary experiments to verify that the quantities of interest easily measurable by off-the-shelf components. Then the validity of our metric and the performance of the proposed algorithm of localizing and identifying people in a crowded scenario are demonstrated by simulations of real experiment data.**

I. Introduction

Networked cameras are increasingly becoming an integral part of many infrastructures for security and surveillance, and several new applications call for their usage in even more places to observe human behaviors and provide services. Furthermore, nowadays many inertial sensors such as accelerometer are equipped into popular mobile devices. Given their widespread availability in this paper we explore a new possibility of localizing a wearable sensors by combining camera observations and accelerometers worn by the people in the camera's field of view. Tracking and recording the position of wearable sensors is one of the most essential information for context-aware computing or life-logging applications [4][10][8].

The core approach in the cooperative localization of wearable sensors and camera is to leverage the linear relationship between a person's walking speed and the standard deviation of their vertical acceleration (bounce), which we verify experimentally. The traditional localization problems tries to find solution of multiple equations representing geometric relationship among nodes in order to track positions of nodes. Instead, our problem tries to find the corresponding accelerometer ID among anonymous path segments from a tracker of camera for the same goal. In practice, however, when people come close together and when they cross paths, the cameras cannot easily disambiguate one from the other. For resolving the ambiguity we propose the path disambiguation algorithm to find the most probable set of path segmentations given accelerometer measurement.

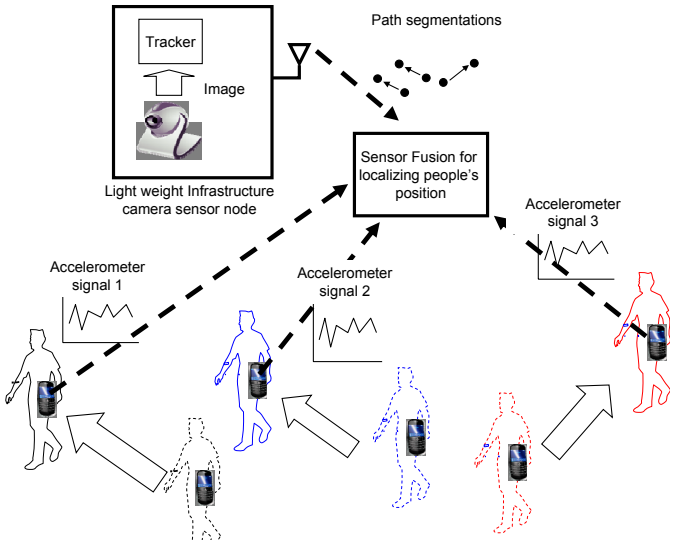


Fig. 1: System Overview

Our system setup is shown in Figure 1. An overhead camera deployed in the infrastructure extracts the centroid positions of people using a background differencing algorithm [11], then a simple tracker generates path segmentations over time by associating a centroid in current frame to one in previous frame. The series of path segments are then correlated with a series of accelerometer measurements transmitted by wireless nodes attached to people's belts to establish correspondence between the unique ID of a person and the silhouette detected by the camera. This gives rise to a new sensing modality where one can use very low end cameras such as the ones used in [11] in conjunction with wireless accelerometers without revealing actual images of the person.

The solution we propose has broad applicability in a wide variety of settings. In Ambient Assisted Living application a person needs to be uniquely identified when making posture measurements in a privacy preserving fashion. In security applications, infrastructures with pre-installed cameras can use the same approach to identify assets and personnel out of a crowd. In service oriented systems, it relaxes the requirement

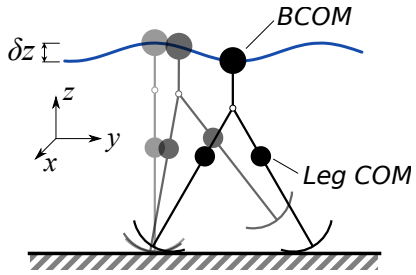


Fig. 2: Inverted pendulum model of human gait. The body center of mass (BCOM) oscillates in the z direction as the person moves forward (y direction).

of complete camera coverage. People can move across very sparse camera setups and still be uniquely identified. The key contribution of this work is to propose the algorithm of localizing uniquely people's position without specialized hardware (e.g. ultrasound) by matching common features between two independently collected signals, one coming from the infrastructure and one from a wearable sensor with the proof-of-concept demonstration.

Our presentation is organized as follows. The second section provides some background on accelerometer sensing with respect to human posture and surveys the related work. Preliminary experiments are conducted to verify the hypothesis we used as a building block in this paper. The third and fourth section describe our matching algorithm for the case when there is no tracking ambiguity followed by the path disambiguation algorithm. In fifth section, we validate our algorithm through experiments and simulations. The last section concludes the paper.

II. Statistical Analysis of Sensor Data

In this section we describes the underlying principles for modeling body movement using accelerometer measurements and how it relates to camera data.

A. Background and linear model formulation

The measurements from both the accelerometer and the camera can be modeled according to the theories of kinetics [2]. Figure 2 illustrates a simplified biped model of walking known as the inverted pendulum model [3], where the legs act as upside-down pendulums attached to the trunk and Body Center Of Mass (BCOM). When humans walk, the most relevant accelerations occur in the y and z axes of the accelerometer. This is because as the person is walking the BCOM oscillates in the "up and down" (z) and "front and back" (y) directions. Therefore an accelerometer attached on the body does not give the actual acceleration data of BCOM. Instead, the accelerometer data describes the body oscillation movement which is converted to work of forward or backward movement [7]. Meanwhile, the displacement (and velocity) of camera centroids describes the BCOM movement very closely.

The statistical model for the vertical acceleration a_z of a moving person can be inferred by analyzing the gait cycle. The maximum and minimum values for the z-acceleration during the gait cycle take place when the foot makes contact with the ground. The time between each consecutive foot-ground contact shows very little vertical acceleration. As the person's speed increases, the stepping frequency is expected to increase, and a larger fraction of the gait cycle is spent in the contact phase. Therefore, the velocity of a human body is closely related to the magnitude of swing in z-acceleration and y-acceleration. Based on the intuition and observation from preliminary experiments we deduce the following linear regression model in (1).

$$v_{BCOM}(k) = \beta_0 + \beta_1 s_{a_z}(k) + \beta_2 s_{a_y}(k) + e_k \qquad (1)$$

where e_k is the zero mean gaussian statistical error, $s_{a_z}(k)$ and $s_{a_y}(k)$ are the standard deviation of z and y-acceleration. Since the two variables are mutually related and measure the same quantity, we use the z oscillation of the BCOM only, i.e $\beta_2 = 0$ in (1).

In the case of camera data, the speed of BCOM can be easily computed from centroid displacement,

$$v_{BCOM}(k) = \sqrt{(x_k - x_{k-1})^2 + (y_k - y_{k-1})^2}/\delta t \qquad (2)$$

where δt is the time between the k^{th} and $k-1^{th}$ frames, and x_k and y_k are the image coordinates of centroid at k^{th} frame.

B. Experiments

To validate the hypothesis that walking speed is proportional to the standard deviation of the vertical component of the measured BCOM acceleration, we designed an experiment that did not make use of a camera, so as to avoid centroid estimation errors, blobbing artifacts, perspective and intrinsic calibration effects. We computed the average walking speed of a person by using a predefined course of known dimensions and measuring the total walking duration. The person wore an accelerometer sensor node attached to the belt, on the front side of their body. A metronome was employed to help maintain a constant pacing frequency. Each experimental run lasted 1 minute, at which point the person stopped and the total walking distance was measured. There were 10 experimental runs in total, using different pacing frequencies. The accelerometer was sampled at 100Hz. Figure 3 plots the standard deviation of the vertical acceleration against the calculated walking speed. The linear trend can be clearly seen in the plot, where a fitted line is shown for comparison. A segment of the time-series for three of these experiments is shown in Figure 4. Here, the hypothesized proportionality between standard deviation and BCOM speed can be clearly observed.

To quantify the effects of camera noise, a similar experiment was performed, but this time the BCOM speed was estimated from camera centroids. The person walked in an unspecified path for five experimental runs and was allowed to walk at different speeds as well as to stop. The centroid was extracted from the image sequence by calculating the center of mass of foreground blobs in the image. Since the experiment was

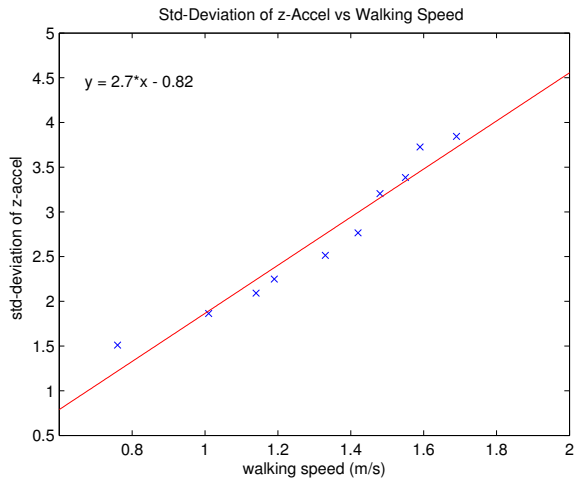


Fig. 3: Measured standard deviation of a_z for people walking at different constant speeds. The data closely follows the linear trend line.

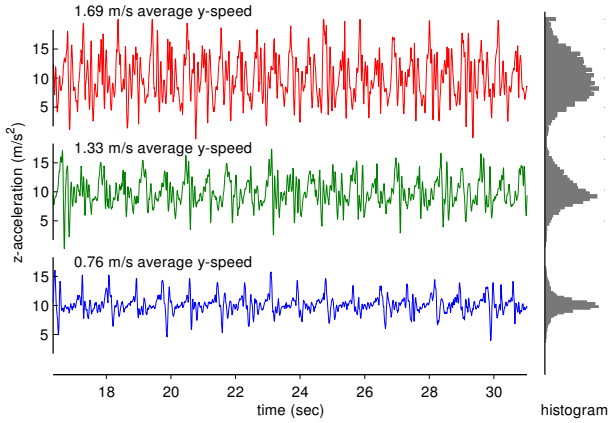


Fig. 4: Vertical acceleration measurement for a person at a constant walking speed of: $1.69m/s$ (top), $1.33m/s$ (middle) and $0.76m/s$ (bottom). The plots on the right show the histogram of acceleration measurements for the whole duration of the experiment (1 minute). As hypothesized, the spread of the distribution varies with speed. This can be explained by observing the top and bottom plots: the low speed signal spends more time in the *swing* part of the gait cycle than on *contact*.

performed in a scene with a static background and with only a single person, the centroid gives a very good estimate of the person's BCOM. Figure 5 shows the outcome of a single run of this experiment, while similar outcomes were found for the four other runs. The hypothesized linearity is reinforced by the experimental data.

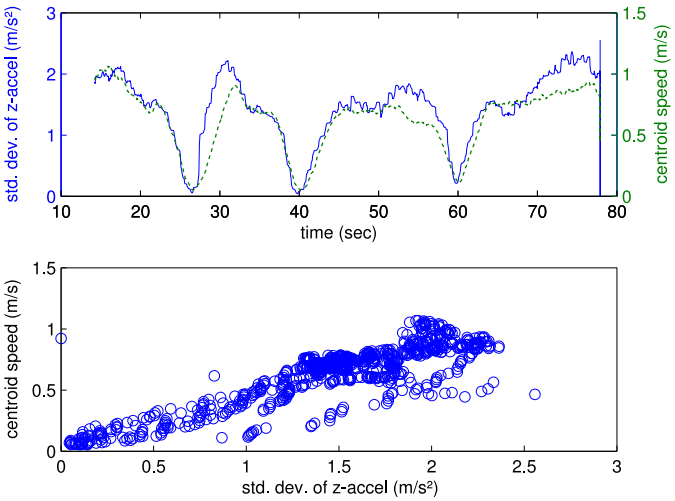


Fig. 5: Top: standard deviation of vertical acceleration (solid) overlaid onto the centroid speed (dashed). Bottom: scatter plot of the standard deviation of z-acceleration versus centroid speed.

III. Similarity Measure Between Accelerometer and Camera Data

We use the correlation coefficient of velocity estimated from sensing data of camera and accelerometer to quantify similarity of those two signals. The correlation coefficient approaches 1 (-1) as two signals are positively (negatively) correlated. If not correlated, it becomes 0. The correlation coefficient of n samples between two discrete signals X and Y is approximated as (3).

$$\rho(X,Y) = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{\sqrt{n\sum {x_i}^2 - \sum {x_i}^2}\sqrt{n\sum {y_i}^2 - \sum {yi}^2}} \quad (3)$$

where x_i and y_i are ith sample of signal X and Y.

The correlation coefficient offers a robust and simple way of quantifying the similarity among signals. In our case, the accelerometer signals are subject to calibration errors mainly caused by either inconsistent orientation with respect to gravity or bias in raw data conversion. The correlation coefficient calculation minimizes the effect of those calibration errors by eliminating dependency on the average value of signal. The magnitude mismatch between the two signals is well canceled out by the average value subtraction in equation (3).

We compute equation (3) in more efficient form using sufficient statistics. Let R_k denote a vector of the sufficient statistics of computing correlation coefficient at time k,

$$R_k = [\sum_{1\le i\le k} x_i \sum_{1\le i\le k} y_i \sum_{1\le i\le k} x_i^2 \sum_{1\le i\le k} y_i^2 \sum_{1\le i\le k} x_i y_i]$$

Then we can compute the correlation coefficient at time k with R_k defining a function, $f_\rho : \mathbf{R}_k \mapsto \rho(X_{1:k}, Y_{1:k})$.

Let $a_{T,m}$ and $c_{T,n}$ denote the acceleration measurement of index m and the camera centroid measurement of index n during the time interval T. The indexes, m and n are

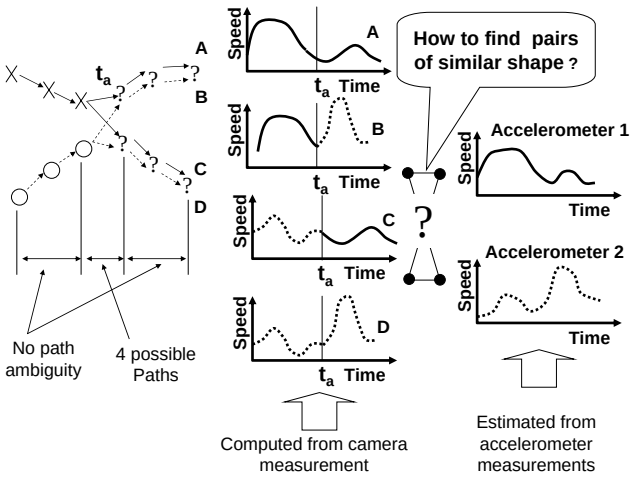


Fig. 6: Problem overview : Two objects are moving in cameras field of view and two accelerometers are used for determining the unique IDs of the objects

the accelerometer node address (ID) and the unique label of centroid trace as assigned by a tracker. Let f_σ denote a function which computes the moving average of the standard deviation of a_z. Similarly, let f_v denote a function that computes the moving average of the centroid velocity. We assume that the function outputs are interpolated with the same sampling rate for correlation coefficient computation if the sampling rates of two signals are different. Then we can describe the function that searches the best matching centroid trace label n^* out of N centroids for mth accelerometer signal during the observation time T as following.

$$n^*(m,T) = \underset{1\le n\le N}{argmax}\ \rho(f_\sigma(a_{T,m}), f_v(c_{T,n})) \tag{4}$$

where ρ represents the correlation function from equation (3).

The function (4) can be directly applied to discern the correct ID assignments when there are no path ambiguities. A tracker, however generates multiple possible labels on centroids when paths cannot be reliably disambiguated by the tracker. For the situations , we propose the path disambiguation algorithm in the next section.

IV. Path Disambiguation

The path ambiguity problem arises when a tracker associates one object with more than two objects in two consecutive image frames. In our system, we assume that the position of people is the only available information from a camera sensor not considering other advanced feature detection algorithms. Our tracker simply constructs the most likely paths by binding the closest centroids between frames. However,it can generate many ambiguous paths when more than two objects come across each other. Figure 6 illustrates the matching problem of signals between accelerometers and a camera under the path ambiguity . In the Figure, two objects, the cross and the circle move in the field of camera view (FOV) area. When two objects come across at time t_a a tracker generates two possible sets of paths, $\{A, D\}$ and $\{B, C\}$.

For determining unique ID of those two objects using two accelerometers 8 correlation computations are required between $\{v_A,\ v_B,\ v_C,\ v_D\}$ and $\{v_{acc1},\ v_{acc2}\}$. The set of velocity traces estimated from a_z (e.g. $\{v_{acc1},\ v_{acc2}\}$) can be used as a reference signal for searching a set of path segments (e.g. $\{A, B, C, D\}$) for particular centroid ID. With the path ambiguity the complexity of matching the measurements from accelerometers and an camera exponentially increases. For k path ambiguities it already leads $O(2^k)$ of computational complexity. To resolve the complexity we developed a disambiguation algorithm. It groups ambiguous paths into clusters based on the pattern of correlation coefficients, then eliminates clusters of the undesirable pattern (e.g. $\{B, C\}$).

For simplicity, in this discussion we treat a tracker as a black box. The tracker receives a randomly permuted set of centroids from a camera and returns ordered arrays of centroids with the array index representing the centroid ID assigned by the tracker. When centroids have multiple competing ID assignments the tracker assigns equal probabilities for each hypothesis. When there is no ambiguity the tracker outputs a single centroid array, and its probability is set to 1.0. If two paths are equally likely the tracker outputs a set of two centroid arrays. The goal of the algorithm is to maximize the number of correct centroid and accelerometer pairs by observing the correlation coefficient value of velocities during the observation time. We model this as a non-linear optimization problem and use a combination of techniques to solve it.

A. Path Disambiguation as Non-linear Optimization Problem

Our problem is to maximize a performance metric defined by the matching rate, i.e. the number of correct matchings over the total number of matchings between accelerometers and centroids. The sorted arrays returned by the tracker represent the possible associations of centroids between the previous and the current frame. These associations can be represented by a permutation matrix θ, which is itself a permutation of the identity matrix I. That is, the elements of θ are in $\{0, 1\}$, and only one 1 can appear per column and per row. Let θ_t denote the permutation matrix from time $t-1$ to t. Let $I_A(x)$ denote an indicator function where $I_A(x) = 1$ if $x \in A$, and $I_A(x) = 0$ otherwise. Furthermore, let $\rho(i, j|H)$ denote the conditional correlation coefficient of ith signal and jth signal under hypothesis H. Assuming N accelerometers and centroids, the matching problem with path ambiguity during time T can be formulated as the following optimization problem. (5).

$$\underset{\theta_1,\ldots\theta_{k_T}}{max}\ \left[\frac{1}{T}E\left\{\frac{1}{N}\sum_{i=1}^{N} I_i(\underset{j\in\{1..N\}}{argmax}\ \rho(i,j|\theta_1,\ldots\theta_{k_T}))\right\}\right] \tag{5}$$

where k_T represents the sample index at time T and $E\{\cdot\}$ is the expectation value.

There exist standard techniques to solve (5). The solution broadly falls into two categories. One is searching a set of state spaces and find deterministic solution. It can be implemented by a class of shortest path algorithm. The other one is estimating the most probable solution based on probability functions. A typical example is Bayesian estimation, maximizing posterior probability density function. The proposed optimization problem in (5), however, implies that the Bayesian estimation solution deals with highly non-linear functions such as f_ρ and non-Gaussian noise. This non-Gaussian nature of the problem makes the Kalman filter and its variants inapplicable.For those reasons, the Bayesian approaches become less attractive than a deterministic solution. The caveat in applying a deterministic solution is that the state space exponentially increases over time. Therefore, it is essential to prune a set of state spaces at a certain point. We find a sub-optimal solution of (5) using a tree pruning algorithm.

B. Cluster Based Tree Pruning

Our search algorithm follows a tree structure where a leaf node represents a hypothesis of path segmentations, $\{\theta_1, ...\theta_{k_T}\}$ up to the current time. It is often computationally impossible to search all the sub-trees since the number of resulting sub-trees grows exponentially with the number of trace ambiguities. Instead, our algorithm finds the best sub-tree of the original tree which is likely to maximize the performance metric in (5). The pruning algorithm consists of three stages. First, the best sub-trees are chosen by evaluating the credibility of the current hypothesis of leaf. Second, using the metric the algorithm clusters the leaf nodes into groups and prunes the subset of groups with lower metric values. In the final stage, it reconstructs traces and the matching sequence once the tree has only one leaf. The detailed pruning procedure is explained below.

Hypothesis Quality Metric Prior to pruning sub-trees we have to evaluate how credible a given path hypothesis is compared to others. For the purpose we introduce the correlation coefficient distance metric. Its conceptual illustration is shown in Figure 7(a). Let e_0 denote a mismatch between an accelerometer and a centroid trace, and let e_1 denote a correct match. Then $P(\rho, e_1|H)$ is a distribution of correlation coefficient of matched signals (mismatched signals in the case of e_0) given that the current hypothesis, H is true. In the figure, the left side of the overlapped area between two distributions represents the error probability of *missing*, p_M, ,and the right side represents the error probability of *false alarm*, p_F. Therefore, we can conclude that the hypothesis, H is more credible if $p_F + p_M$ (the probability of incorrect matching between two signals given a hypothesis, H)is smaller over time in Figure 7(a). The simplest way of gauging $p_F + p_M$ is to measure how far those two distributions are separated each other by which the overlapped area become smaller. Based on this observation we propose the correlation coefficient distance metric given H, shown in (6).

$$D(\rho|H) = |E(\rho, e_0|H) - E(\rho, e_1|H)| \tag{6}$$

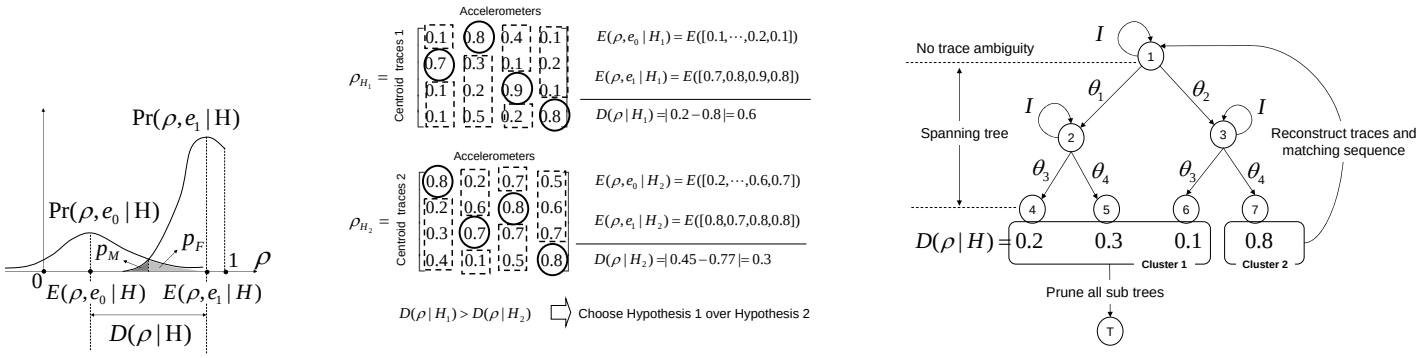
In (6), the correlation coefficient distance is computed by the absolute difference of average correlation coefficient between estimated non-matching and matching signals assuming the hypothesis, H is true. An computation example of (6) is presented in Figure 7(b). In the figure, the thick circles and dotted rectangular represent the matched and non-matched signal pair respectively. In the example, two path segmentation hypothesis generates two different correlation coefficient matrices, and the first hypothesis is chosen since its correlation coefficient distance is greater than the other.

Leaf Clustering and Pruning Leaf nodes sharing the same parent (or ancestor) nodes tend to converge to the same correlation coefficient distance since they have a common set of path segments. This observation leads to a heuristic for pruning leaf nodes by clustering leaf nodes with similar correlation coefficient distances. We use an agglomerative hierarchical clustering algorithm [12] due to its low time complexity. The cluster distance represented by $\|\cdot\|$ is defined by the smallest correlation coefficient distance between leaf nodes in the two clusters (i.e. single linkage). The aggregated false associations of path segments results in outlier clusters of leaf nodes. The pruning algorithm detects and prunes all leaf nodes in those outlier clusters. We use a simple absolute threshold value for detecting the outlier cluster. Let $c_i(t)$ denote ith cluster at time t where $i \in U_t = \{1...u_t\}$ and u_t is the number of clusters at time t. Then outlier clusters, ω_t are defined as :

$$\omega_t = \{\ i \mid \|c_j(t) - c_i(t)\| \geq \lambda,\ j = argmax_{k \in U_t}(\|c_k(t)\|)\} \tag{7}$$

where the λ is the threshold value of distance between outlier cluster and the others. In (7) we define outlier clusters if their cluster distance from jth cluster is greater or equal to λ where jth cluster has the largest correlation coefficient distance value. Figure (c) illustrates the leaf crusting and pruning process. In the figure four possible path associations (node 4,5,6 and 7) are shown, and the algorithm constructs two clusters, cluster 1 with [0.2 0.3 0.1] and cluster 2 with [0.8]. Then node 4, 5, and 6 are classified outlier leaf nodes and pruned if $\lambda \leq 0.5$ since the cluster distance is smaller than λ.

A small value of λ reduces the time complexity of pruning algorithm, but also degrades the matching performance. Therefore, the optimal value should be chosen in determining desirable trade-off between those two conflicting goals. In order to quantify the trade-off, we develop a cost function of λ. Let T_p denote the average processing time per frame of the disambiguation algorithm and F_r denote the frame rate. The time complexity cost is formulated by the service rate, $U_R(\lambda) = F_r T_p(\lambda)$, the proportion (in percentage) of processing time over the inter-arrival time of image frames. The quantity is often introduced as the utilization factor [1] in queuing theory. We note that processing delay in the



(a) (b) (c)

Fig. 7: (a) Correlation coefficient distance measure; (b) Illustrative example of computing the correlation coefficient distance; (c)Example of tree pruning algorithm;

system could indefinitely grow over time if $U_R > 100$. The performance cost is simply defined by matching error rate. The cost function places equal weight on both quantities. The proposed cost function is shown in (8).

$$cost(\lambda) = 100U_R(\lambda) + (100 - M_R(\lambda)) \tag{8}$$

where M_R is the matching rate.

Reconstructing Trace and Matching Sequence We can improve the matching rate by tracing back the path hypothesis if we relax the real-time computation requirement. Let l_t denote the number of leaf nodes at time t, i.e. $l_t = \sum_{i=1}^{u_t} |C_i(t)|$. The path ambiguity is resolved at time t when only one leaf node is left after the pruning process ($l_t = 1$). Then the tree reconstructs path traces of centroids with matching sequences since the matching IDs of centroids can be recursively computed in the tree once the matching IDs of centroid are uniquely determined at time t.

V. Performance Evaluation

The proposed algorithm is validated in 3 steps. First, we validate the similarity metric in (4) using accelerometers and camera data set collected by 12 independent experiments. Second, we present an extensive evaluation of path disambiguation algorithm discussed in section IV through computer simulations using the experiment data set. In our experiment system a ceiling-mounted camera with Intel iMote2 nodes[9] captures images and computes the centroid position of a person. Since the experiments consist of capturing data for a single person at a time, we program the iMote2 nodes with a single-person detection algorithm (as used in [6]) which calculates the center of mass (centroid) of all foreground pixels. This avoids segmentation artifacts, reducing centroid location noise. The cameras are mounted on the ceiling, facing down, in order to provide a good approximation of the person's floor-plane position. We recorded 12 sets of centroid traces and accelerometer measurements via 12 independent experiments. In each experiment a person walks for 1 minute in a $4 \times 5m$ space one at a time . An iMote2 camera node is installed on a 12-foot height ceiling, outputting the person's centroid 15 times per second. The wearable sensor node fitted with an Analog Devices ADXL330 accelerometer is attached to the person's waist. The node transmits its measurements to a computer via a Zigbee wireless link. For these experiments, a sampling period of $70ms$ is used. We note that the movement is performed in unplanned and random manner including non-walking activity such as jumping, sitting, running, lifting legs etc.

A. Similarity Measure Performance

We use the 12 sets of walking traces shown to verify the similarity measure, but the ID of the accelerometer measurements corresponding to a given centroid trace, however, remains unknown. We compute the proposed similarity measure in equation (3) for all pairs of accelerometers and centroid traces. The correlation coefficient result is shown in Figure 8. The accelerometer sensor with the highest correlation coefficient is selected as the best matched one out of the 12 traces in the figure. As shown in the plot, all accelerometers are correctly matched with camera centroid traces, verifying the validity of our correlation choice.

B. Path Disambiguation Performance

For this experiment we assume that multiple people walk in camera field of view and tracker often gives incorrect traces of centroids. The experiment is designed with a mixture of the 12 experiment data sets and MATLAB simulation. In that way, we can exclude other error sources such as multiple people detection error and focus on the errors caused by centroids crossing. We integrated the 12 experimental data sets into one time reference by linear interpolation of 20 samples per second. The scenarios of n people walking are created by randomly selecting n data set out of 12. Therefore, we have $\binom{12}{n}$ data sets for each n person scenario. All results are obtained from 50 random sample out of $\binom{12}{n}$. The trace IDs of

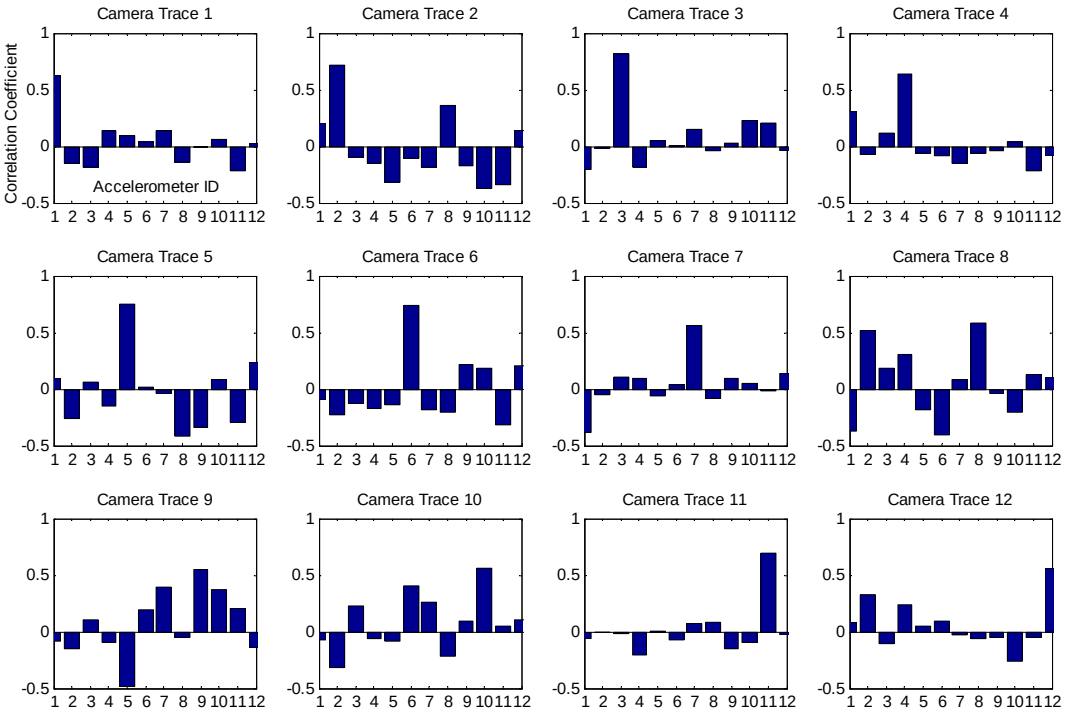


Fig. 8: Correlation coefficient result for the case where there is no tracking ambiguity: i^{th} each trace and i^{th} node have the maximum correlation coefficient

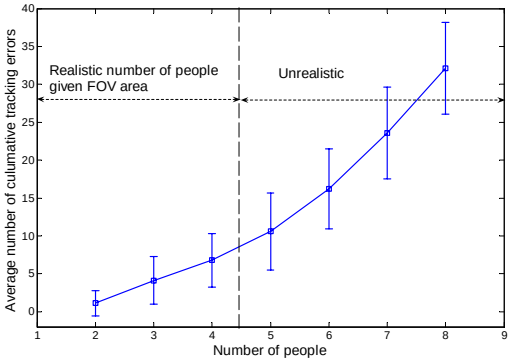


Fig. 9: Tracker performance

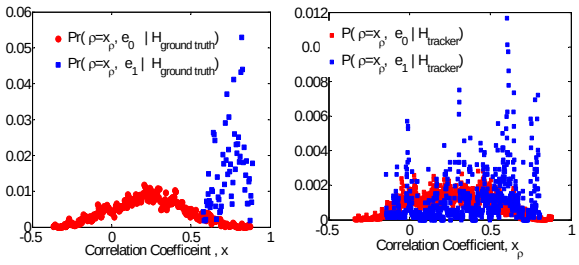
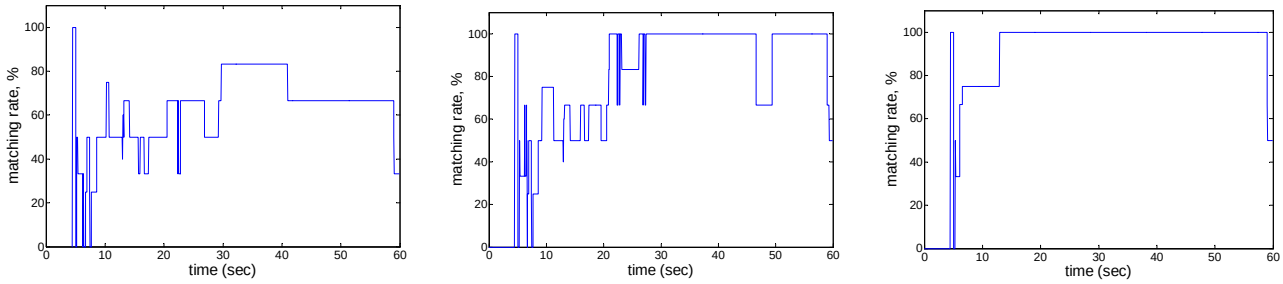


Fig. 10: Sampling distribution of the correlation between matched measurements (squares) and discarded matchings (circles) when matched measurements are correct (i.e. correspond to ground-truth matchings)

centroids are blinded by a random permutation of n positions, and then the permuted centroid data is sent to a tracker.

We implemented a simple tracking algorithm by associating the closest centroids between frames. The tracker generates multiple possible associations given a path ambiguity, i.e. more than two centroids are overlapped. Figure 9 shows the performance of the implemented tracker, i.e the number of wrong association over number of people. We note that although more than 4 people walking in the given area (whose size is $4 \times 5m$) is unrealistic, it is included in the experiment in order to examine the limiting performance of our system. The number of tracking errors grows with polynomial order as the number of people increases as shown in the figure. We use the number of people for the baseline experiment control parameter instead of the number of tracking errors because its quantity is more intuitive. The key performance metric is a percentage of centroids with correct matching ID at a given time, i.e. the objective function in (5).

Correlation Coefficient Distance We verify our argument on the correlation coefficient distance by analyzing the experimental data. We compute the correlation coefficient matrix with ground truth, $H_{ground\ truth}$ and with hypothesis output of tracker output, $H_{tracker}$, then compute $p(\rho, e_0|H)$



(a) (b) (c)

Fig. 12: Matching rate comparison for 6 people walking scenario a) tracker only, b) path disambiguation algorithm without reconstruction, c) path disambiguation algorithm with reconstruction

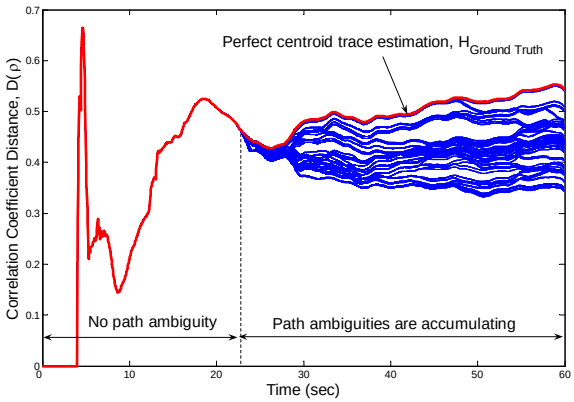


Fig. 11: $D(\rho, e|H)$ trace of leaf nodes in tree for 6 people walking scenario

and $p(\rho, e_1|H)$ for 12 people walking data set. We note that the tracker gives 62 trace errors in total for 1 minutes for 12 people data set. As shown in Figure 10, the distance between $p(\rho, e_0|H)$ and $p(\rho, e_1|H)$ with $H_{ground\ truth}$ is significantly larger than with $H_{tracker}$. Furthermore, we observe the correlation coefficient distance trend of 568 leaf nodes in tree for 60 seconds given 6 people walking scenario as shown in figure 11. The red thick line is the correlation coefficient distance given that all path traces are perfectly estimated, i.e $H_{ground\ truth}$. In the figure, the path ambiguities start at 23 seconds. As predicted, the distance metrics form groups centered in a certain value and those groups diverge from each other over time. A leaf node with $H_{ground\ truth}$ maintains the highest correlation coefficient distance among other leaf nodes.

Performance over disambiguation stages The algorithm performance is evaluated by the matching rate with $\lambda = 0.05$. Figure 12 shows the matching rate over time for different configurations given a 6 people walking case. It shows that the matching rate performance significantly improves through the proposed disambiguation algorithm. In the first figure, the matching IDs are directly obtained from correlation coefficient without disambiguation process where a path hypothesis is randomly chosen from the tracker. The second and third figures show the matching rate when the disambiguation algorithm is used without (Figure 12b) and with matching sequence reconstruction (Figure 12c). In Figure 12b the matching IDs are generated at each time from the leaf node with the best correlation coefficient distance among all leaf nodes and the previous matching sequences are not re-labeled, i.e. instantaneous matching rate. Meanwhile, the previous matching sequences are re-labeled by reconstruction in Figure 12c. In this figure, the first person enters camera view at 5 seconds. We note that it takes at least 5-10 seconds in order to compute meaningful velocity values. The first 5-10 second of unstable matching rate is explained by the velocity convergence time. As can be seen, the disambiguation algorithm correctly finds all matching IDs of centroids after 14 seconds.

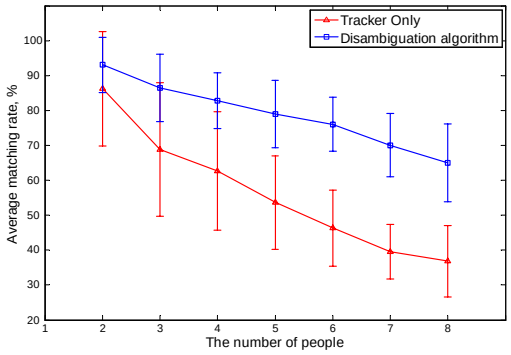


Fig. 13: Average matching rate

Performance over complexity of scenario In this experiment, we compare the matching performance between tracker-only and disambiguation algorithm with $\lambda = 0.05$ as we increase the trace complexity, i.e. the number of persons. As shown in Figure 13, the performance gap is widening as the number of persons increases. The performance becomes twice in the 8 people walking scenario. The matching performance, however is obtained with the large processing

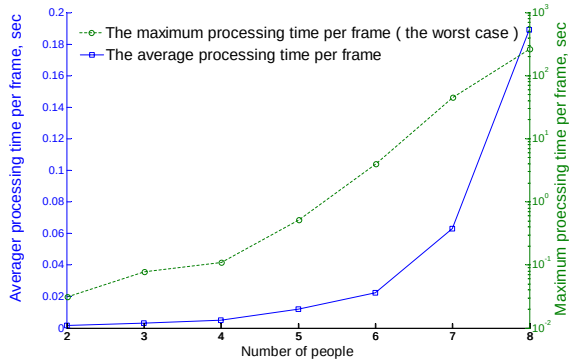


Fig. 14: Processing time

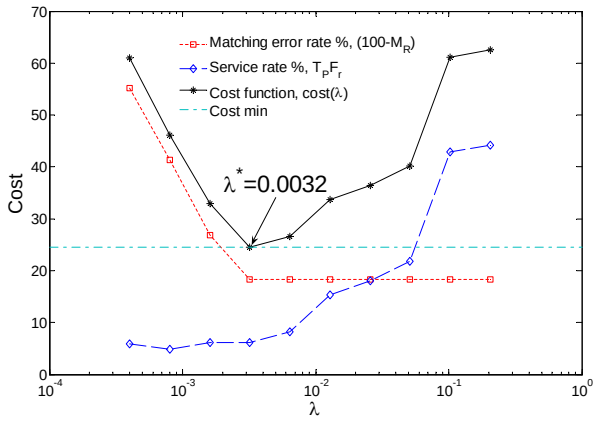


Fig. 15: Cost function trend over λ given 6 sample traces randomly chosen from trace set in Figure (**??**), Traces set =$\{1,\ 2,\ 5,\ 6,\ 7,\ 9\}$

time cost as shown in Figure 14. The average processing time, T_p exponentially grows to 0.01, 0.02, 0.06, 0.2 second from 4 persons to 8 persons complexity. Specially, with 7 people complexity the service rate exceeds 100%, $U_R = 125\%(= 100 \times 0.0625/0.05)$. Therefore, the processing time could indefinitely grow in the worst case as shown in the maximum processing time. Such a large processing time cost can be significantly reduced by selecting the proper value of λ using the cost function in (8).

Cost function over λ In the previous experiment, the non-optimal value of λ causes a large processing workload. The workload, however, can be minimized by choosing the optimal λ while maintaining the same matching rate performance as shown in Figure 15. In the figure, matching error rate, service rate, and cost function are drawn over different $\lambda = \{2^k \cdot 10^{-4}\}$ values where $k = 2, 3 \ldots 11$ for the 6 people walking scenario randomly sampled from the trace set in Figure (**??**). The figure shows the clear trade-off between the service rate (or processing time) and the matching error rate over λ. The cost function is minimized at (processing time: 0.003 sec, matching rate: 81.63%). Comparing to the matching rate 82% and the average processing time, 0.02 sec in Figure 13 and 14 the optimal λ reduces processing time more than 6 times with relatively the same matching rate.

VI. Conclusions

In this paper we introduced a new approach of localizing wearable sensors using sensor fusion modality of wearable accelerometer measurements with people detections made by the infrastructure camera. Our experiments have shown that the proposed disambiguation algorithm operates reliably, degrading gracefully even when people presence in the scene becomes too dense. Our proposed algorithm specially have a great potential impact on the application where a system needs to consistently identify people for the long period. The constraint of accelerometer position (waist) can be relaxed by compensating the tilt of body using additional inertial measurement sensors such as gyroscope. The key feature of our proposed algorithm is to use well-defined metrics and key statistics for minimizing data size and computation load. The computation complexity and matching performance can be optimally compromised by the control knob of λ. Further improvements of the disambiguation algorithm and system design issues related to the energy and wireless network will be the topic of our future work.

References

[1] D. Bertsekas and R. Gallager. *Data networks*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1987.

[2] S. A. Gard, S. C. Miff, and A. D. Kuo. Comparison of kinematic and kinetic methods for computing the vertical motion of the body center of mass during walking. In *Human Movement Science*, volume 22, pages 597–610, 2004.

[3] A. D. Kuo. The six determinants of gait and the inverted pendulum analogy : A dynamic walking perspective. In *Human Movement Science*, volume 26, pages 617–656, 2007.

[4] M. L. Lee and A. K. Dey. Lifelogging memory appliance for people with episodic memory impairment. In *UbiComp '08: Proceedings of the 10th international conference on Ubiquitous computing*, pages 44–53, New York, NY, USA, 2008. ACM.

[5] D. Lymberopoulos and A. Savvides. Xyz: A motion-enabled, power aware sensor node platform for distributed sensor network applications. In *IPSN, SPOTS track*, April 2005.

[6] D. Lymberopoulos, T. Teixeira, and A. Savvides. Detecting patterns for assisted living: A case study. In *Proceedings of SensorComm*, 2007.

[7] W. E. McIlroy and B. E. Maki. The control of lateral stability during rapid stepping reactions evoked by antero-posterior perturbation: does anticipatory control play a role? In *Gait and Posture*, volume 9, pages 190–198, 1999.

[8] A. Meschtscherjakov, W. Reitberger, M. Lankes, and M. Tscheligi. Enhanced shopping: a dynamic map in a retail store. In *UbiComp '08: Proceedings of the 10th international conference on Ubiquitous computing*, pages 336–339, New York, NY, USA, 2008. ACM.

[9] L. Nachman. Imote2, http://www.tinyos.net/ttx-02-2005/platforms/ttx05-imote2.ppt, 2006.

[10] D. H. Nguyen, A. Kobsa, and G. R. Hayes. An empirical investigation of concerns of everyday tracking and recording technologies. In *UbiComp '08: Proceedings of the 10th international conference on Ubiquitous computing*, pages 182–191, New York, NY, USA, 2008. ACM.

[11] T. Teixeira and A. Savvides. Lightweight people counting and localizing in indoor spaces using camera sensor nodes. In *ACM/IEEE International Conference on Distributed Smart Cameras*, September 2007.

[12] H. Trevor, T. Robert, and F. Jerome. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, August 2001.