

## **Automatic calibration of laser range finder positions for pedestrian tracking based on social group detections**

DYLAN F. GLAS, FLORENT FERRERI, TAKAHIRO MIYASHITA,  
HIROSHI ISHIGURO, and NORIHIRO HAGITA

*[dylan, florent, miyasita, hagita]@atr.jp, ishiguro@sys.es.osaka-u.ac.jp*

### **Abstract**

Laser range finders (LRF's) are non-invasive sensors which can be used for high-precision, anonymous tracking of pedestrians in social environments. Such sensor networks can be used in robotics to assist in navigation and human-robot interaction. Typically, multiple LRF's are used together for such tasks, and the relative positions of these sensors must be precisely calibrated. We propose a technique for estimating relative LRF positions by using observations of social groups in the pedestrian flow as keypoint features for determining coarse estimates of relative sensor offsets. The most likely offset is estimated using a generalized Hough transform and used to identify sets of possible shared observations of individual pedestrians between pairs of sensors. Outliers are rejected using the RANSAC technique, and the resulting shared observations from each sensor pair are combined into a constraint matrix for the sensor network, which is solved using least-squares minimization. Results show calibration accuracy of sensor positions within 34mm and 0.51 degrees, and an analysis of pedestrian data collected from ubiquitous networks in three public and commercial spaces shows that the proposed calibration technique enables pedestrian tracking within 11 cm accuracy.

*Keywords: Sensor calibration, laser range finders, pedestrian tracking, social groups, ubiquitous sensing*

## **1. INTRODUCTION**

In recent years, laser range finders (LRF's) have enjoyed great popularity as a sensor of choice in ubiquitous networks for the tracking of pedestrian motion in social spaces. Research groups around the world have used LRF-based tracking to study human motion [1-4] and in conjunction with robots, to enable safe human-robot interaction in populated environments [5-7].

For pedestrian tracking, laser range finders offer many advantages over other types of sensors. Their non-invasiveness is a great advantage; installing hardware such as floor pressure sensors can be disruptive to public and commercial spaces, and requiring people to carry tags or handheld devices often requires active intervention in the social system being studied. While video is sometimes used as

a tracking tool, LRF's provide much higher measurement accuracy and require far less data processing. Additionally, LRFs output only anonymous range values, presenting less of a privacy concern than video cameras. While these benefits must be balanced against the cost of the sensors, they remain a popular tool for analyzing human motion in high-traffic public spaces.

LRF's have often been used for human tracking, such as in the work by Fod et al., which combines data from multiple LRF's [1], the work by Arras et al., which incorporates machine learning to attain high-precision leg-tracking [8], and the work by Xavier et al., who developed fast techniques for leg detection from LRF data [9].

Our lab in particular has used an LRF-based human tracking system in several field trials, shown in Fig. 1, and we have built a substantial infrastructure on top of this system. We rely on pedestrian tracking to analyze the use of social spaces, to make predictions about human motion [10], to support robot localization [11], to locate specific individuals [12], and to plan robot trajectories to approach or avoid people [13].



Figure 1. Example of sensor network deployed in a shopping area. Sensor poles are placed unobtrusively against walls and columns around the space to provide ubiquitous tracking.

It is thus of critical importance that the human tracking system provide consistently accurate data, for which proper calibration of sensor positions is essential. As our experiments have grown in size and number of sensors, the task of calibration has become more critical to data integrity and more difficult to perform by hand. In this paper we present a technique for calibrating sensor positions automatically, using pedestrian trajectories in the environment. Not only does this reduce the effort required for calibration, such as physically placing landmarks in the environment and manually aligning sensors, but it also achieves calibration non-invasively, i.e., without disturbing the social dynamics of the environment being observed.

The techniques presented in this paper have been tested with our own system. Our configuration may differ from other tracking systems in various ways, such as the use of torso-height sensors as opposed to the more common leg-height sensors, but the solutions proposed in this paper should be

applicable to other tracking systems as well.

## 2. SENSOR LOCALIZATION

The estimation of sensor positions is a common task across many application fields, and there are several related techniques in existing literature. For example, Senior et al. developed a visual technique for automatic camera calibration [14]. Reference [15] provides a survey of available techniques for sensor localization. Regarding LRF position calibration, much existing work is related to robot localization and mapping.

### 2.1. Requirements and constraints

#### 2.1.1. Basic requirements

We are considering sensors in fixed positions, and it is assumed that a global scan map is not available. Due to the nature of laser range finders, it is not possible for the sensors to directly detect the relative locations of other sensors. The only available data is the current and historical range scan data from each sensor. Scan-matching is often used in robot localization techniques such as Monte-Carlo Localization [16] and Simultaneous Localization and Mapping (SLAM) [17], but it cannot be used for stationary sensors because ambiguities cannot be resolved by moving the sensors, a point discussed in more depth in [18].

The calibration procedure should also be non-invasive, enabling localization without interfering in the social environment being observed. Placing large objects in a busy shopping area, for example, could impede the flow of customers or deliveries, obstruct product displays, or disrupt the mood or atmosphere that the business is trying to cultivate. Using landmarks naturally found in the environment also makes it possible to recalibrate the system quickly at any time with minimal effort, e.g., if a sensor were moved while data collection was in progress.

For these reasons, our proposed technique uses pedestrians moving through the environment as features for calibrating sensor positions. Similar work in multi-sensor localization using pedestrians as reference features has been performed with omnidirectional cameras [19], although these techniques cannot be directly applied due to fundamental differences in the nature of LRF and video data.

#### 2.1.2. Extensions from previous work

In previous work, we proposed a method for automatic sensor calibration based on pedestrian observations, where observations were matched between sensors by comparing trajectory shapes [18]. We found that technique to be effective in environments where walking patterns vary significantly, such as within a room with several local destinations. However, when we tried to apply that technique in long corridors and large social spaces where people have distant goals, we found it to be ineffective. In these environments, the majority of trajectories locally approximate straight lines,

making it impossible to discriminate between them based on trajectory shape alone. Thus, one new requirement for our algorithm is to identify discriminatory features other than trajectory shape which will enable reliable matching of observations of the same people from disparate sensors.

Furthermore, the large number of people observed simultaneously in these larger spaces required heavy computation. Our previous approach was brute-force, comparing every possible combination of trajectories between every pair of sensors. These computations become quite heavy for large spaces, as the number of comparisons necessary for  $s$  sensors with  $p$  pedestrian observations each grows as  $ps^2$ . To extend our algorithm to be effective in large spaces with many sensors, a more efficient and scalable approach is necessary. The procedure of matching observations between sensors should be made as lightweight as possible, and minimal time should be spent evaluating unlikely matches.

## 2.2. Proposed solution

### 2.2.1. Patterns in pedestrian motion

Considering these constraints, there are two problems to be solved: First, when most pedestrian trajectories are similar in geometry, we need some way to identify shared observations between sensors. Second, when the number of pedestrians is large, we need a way to reduce the search space to lower the computational load. To address these two points, we propose the use of emergent patterns in pedestrian motion as keypoints, rather than using the pedestrians themselves.

Pedestrian motion through social spaces is quite rich, with a variety of observable features. While the simplest models treat the flow of pedestrians like a fluid, reacting to physical obstructions by flowing around them [20], more detailed models recognize not only physical forces, but social forces as well, which can be observed in crowd dynamics [21, 22]. Some features that could be observed in a flow could be physical, e.g. people slowing down to go through a bottleneck in a corridor; social, for example, people adjusting their paths to avoid an injured or elderly person moving slowly; or psychological, such as people slowing down to look at an interesting shop display. Any of these dynamic events can be observed not simply as independent behaviors of individuals, but as emergent patterns that arise in the pedestrian flow. When such patterns can be observed from multiple viewpoints, they become candidate features which can be used in calibration of sensor positions.

### 2.2.2. Social groups

Although many of these social flow features are highly localized or infrequent, one social feature that is commonly observed in many environments is the *social group*. These groups are formed when pedestrians with some social connection are walking together. Regardless of whether a group consists of a parent and child, a group of friends, a romantic couple, or a team of business colleagues, it can be recognized by the proximity between its members and the fact that members of a group regulate their speed in order to stay together, regardless of external forces on the pedestrian flow. Fig. 2 shows

examples of social groups observed by our tracking system.

Social groups are easy to detect and quite common in many environments. Studies in crowd dynamics have revealed social groups to constitute up to 70% of pedestrian traffic [23]. Thus, we propose to use the social groups naturally formed by pedestrians as pose-invariant features for matching observations between sensors.

Using social groups rather than individual trajectories as feature vectors for matching observations between sensors provides a number of benefits in terms of efficiency and robustness to noise. First, groups have more distinguishing features than individual trajectories, reducing the possibility of false matches. Second, social groups are stable over time, providing opportunities for filtering time-series data for better noise rejection. Finally, social groups are easily detectable based on instantaneous data within a single time-frame, a computationally simpler procedure than full trajectory matching.

### 3. CALIBRATION ALGORITHM

Our algorithm consists of a sequence of steps which are somewhat analogous to the Scale-Invariant Feature Transform (SIFT) [24] and a variety of other algorithms for computer vision which use rotation-invariant features to identify coordinate transformations. The sequence of steps is presented in Table 1, and each step will be explained in this section.

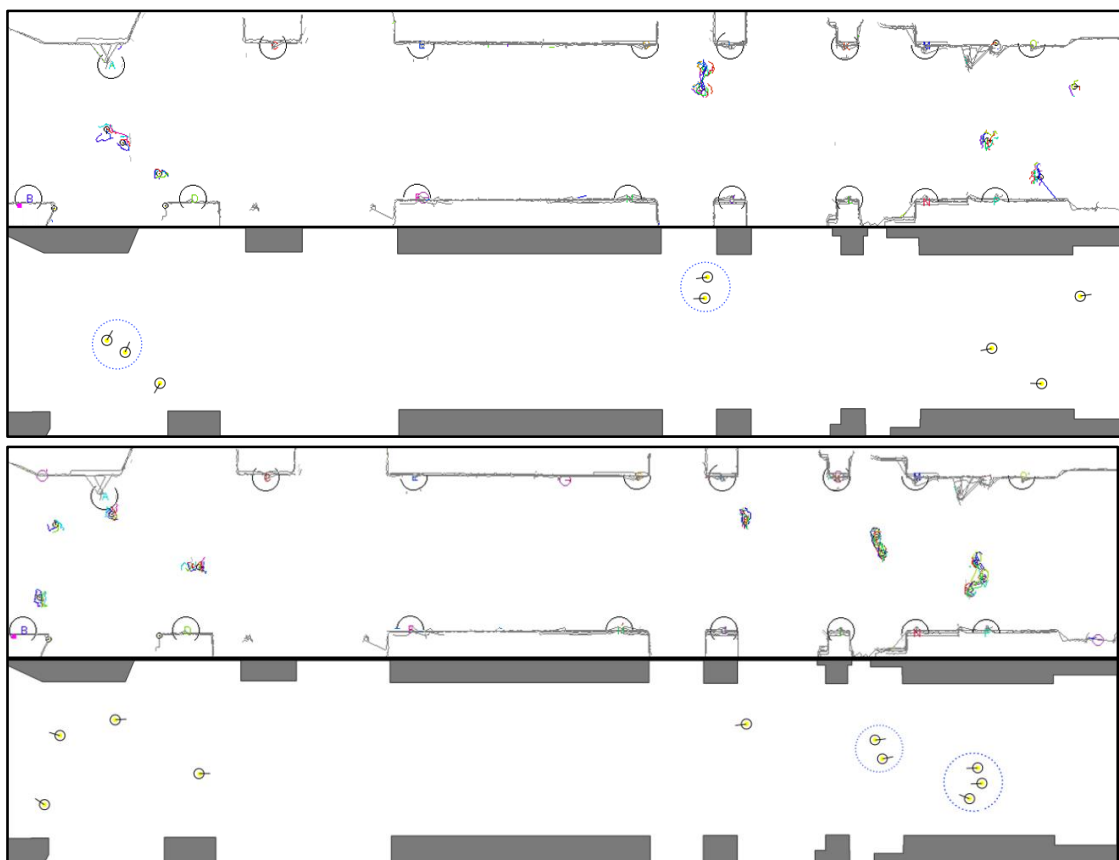


Figure 2. Point cloud and simplified tracking data from two scenes showing social group detections.

**Table 1.** Steps in the proposed calibration algorithm.

<b>Step in Algorithm</b>	<b>Description</b>	<b>Output</b>
Extract human positions (Sec. 3.1)	Using raw scan data and a background model from each sensor, identify pedestrian positions	Sensor-relative pedestrian locations (typically noisy)
Identify social groups (Sec. 3.2)	Based on proximity and coherence of motion direction, identify social groups seen by each sensor	List of groups for each sensor
Compare groups between sensors (Sec. 3.3)	Normalizing groups by motion direction, compare relative positions of members to identify potential matches	List of matching groups for each sensor pair, and a sensor offset hypothesis for each matching group
Generalized Hough transform (Sec. 3.4)	Accumulate sensor offset hypotheses in bins, and select the bin with the highest score	Approximate sensor offset hypothesis for each sensor pair
RANSAC (outlier rejection) (Sec. 3.5)	Perform RANSAC to reject false nearest-neighbor matches and calculate a refined offset hypothesis for each sensor pair.	Pairs of shared pedestrian observations from entire observation history corresponding to the best-fit offset hypothesis
Solution of network (Sec. 3.6)	Build constraint matrix incorporating human observation pairs for all sensor pairs. Solve matrix using least-squares minimization.	Location and orientation for each sensor in the network

### 3.1. Identifying human detections

#### 3.1.1. Build a background scan for each sensor

A background scan is built to model the fixed parts of the environment. Over several scans, a set of observed distances are collected for each scan angle, and the most frequently observed distances over time are used to build the background model. This technique allows us to filter out moving objects like people walking through the area.

### 3.1.2. Extract human positions from scan data

Each data scan is then segmented to extract human positions. Since our system uses torso-level scanning, this is a relatively simple process. Fig. 3 shows examples of real scan data.

A median filter is used to remove outliers and smooth the scan data. Continuous segments of foreground points more than 30 cm in width are then extracted. Discontinuities of more than 30 cm in depth are considered as segment boundaries. Possible occlusions are also considered.

Small gaps are removed between segments of similar distance, and segments between 5 cm and 80 cm in width are considered as human candidates. Using an elliptical shape model (major axis=55cm, minor axis=25cm), a body center estimate is determined as a function of visible body width.

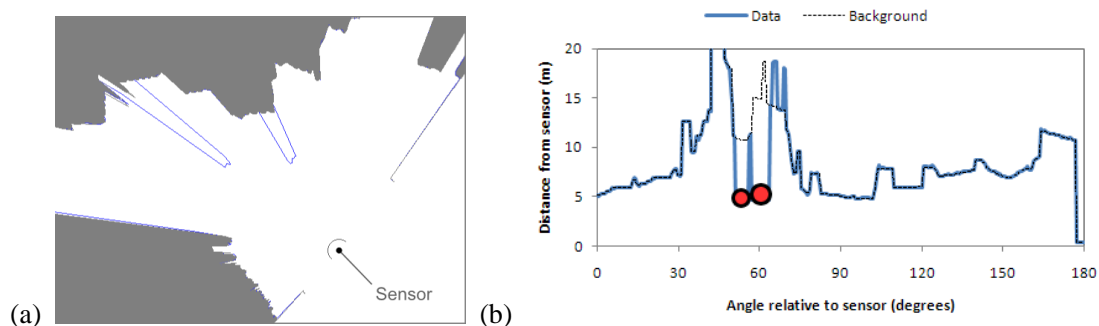


Figure 3. (a) Scan data for two pedestrians in Cartesian coordinates. Gray represents the background model. (b) Polar representation of scan data a few seconds later, showing human detections.

This step outputs a list of relative human positions, in the sensor's local coordinate system.

### 3.1.3. Build trajectories

Using nearest-neighbor matching to the detections from the previous time frame, the human detections are linked into trajectories. These are used for generating stable velocity estimates by time-series filtering, and to ensure observations of the same people at different times are not used to generate duplicate sensor offset hypotheses.

## 3.2. Pose-invariant feature detection

For keypoint features in computer vision to be robustly detectable despite noise and visual transformations, a descriptor vector is defined for each transformation-invariant feature, describing properties which can be used to estimate feature matches between frames.

In our problem space, we propose *social groups* in the pedestrian flow as pose-invariant keypoint features. Social groups afford more rich feature descriptors than individual pedestrian trajectories, and they are stable over time, so noise rejection is possible by time-series filtering.

### 3.2.1. Social groups as pose-invariant features

To use social groups as pose-invariant features, we define a vector of measurable properties which will be consistent when viewed by sensors in different positions. This descriptor vector includes the number of members  $n$  of the group, the magnitude of its motion vector,  $\|\vec{v}\|$  (note that  $\vec{v}$  is defined in sensor-relative coordinates, so only its magnitude can be used as a pose-invariant property), and the geometric description of the group shape.

The shape of the group can be described in a pose-invariant way as a list of the positions of its members. This information is stored as a collection of vectors  $M = \{\vec{m}_1, \vec{m}_2, \dots, \vec{m}_n\}$  describing the position of each member in a coordinate system centered on the group's geometric center and oriented with its x-axis in the group's direction of motion.

### 3.2.2. Detection of social groups

A growing body of research is concerned with the study of social groups in pedestrian motion dynamics. In a study of 1020 pedestrian groups in an urban environment, Costa identified common group formations and studied their associations with gender and social factors [25]. Moussaïd et al. studied the shapes of approximately 1500 social groups and analyzed variations in average interpersonal distances [23].

Techniques for group detection are an active topic of research, e.g. [26]. In this study, we identify pedestrians with interpersonal distances below 1.5 m and a coherent direction of motion (within  $\pm 30$  degrees) as social groups. As our algorithm is robust to noise, this simple definition is sufficient even if it produces some false detections.

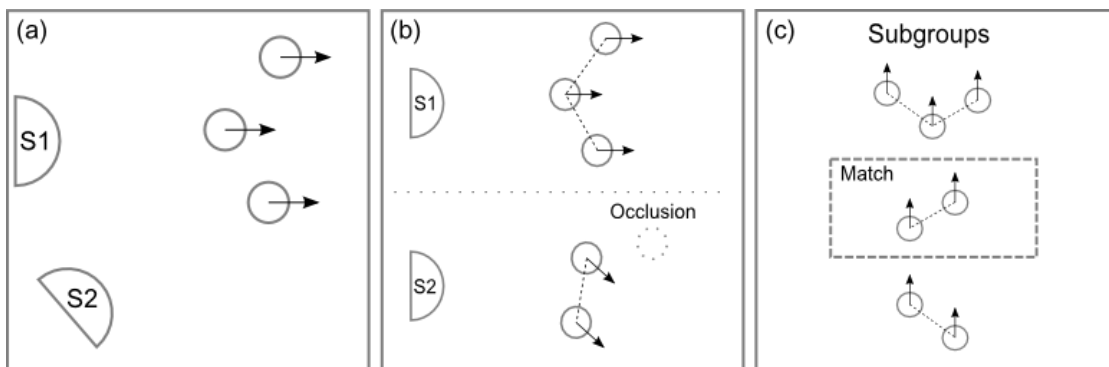


Figure 4. Diagram of a group, showing subgroups and possible occlusions. (a) Sensors and pedestrians shown in absolute coordinates. (b) Sensor-relative observations of the group. (c) Subgroups enumerated, with the subgroup which matches between S1 and S2 highlighted.

### 3.2.3. Enumerate sub-groups to guard against occlusion

Some members of a group may be occluded by others. Thus, a group may appear to have three members to one sensor, but only two to another. To address this possibility, we enumerate all



sub-groups of an observed group to use as candidates for matching between sensors.

Fig. 4 (a) illustrates a 3-person group observed by two sensors, S1 and S2, where one member is occluded from the perspective of S2, as shown in Fig. 4 (b). By enumerating the three possible subgroups, that is, one 3-person group and two 2-person groups, it is still possible to identify a match between the group observed by S2 and a subgroup of the group observed by S1, as in Fig. 4 (c).

### 3.3. Hypothesis generation and feature matching

#### 3.3.1. Comparing groups

As described above, a group descriptor consists of the number of members  $n$ , the magnitude of its motion vector  $\|\vec{v}\|$ , and a collection of member vectors  $M$ . To estimate the likelihood of a match between two groups, we first consider only groups of identical size, that is, where  $n_1 = n_2$ . Optionally, filtering can also be performed based on group speed, that is, where  $|\|\vec{v}_1\| - \|\vec{v}_2\|| < v_{threshold}$ .

Finally, to compare the member vectors, it is necessary to assign correspondences between the members of the two groups. A simple approach is to enumerate members counterclockwise around the group center, beginning from the  $x$ -axis, as in Fig. 5 (a).

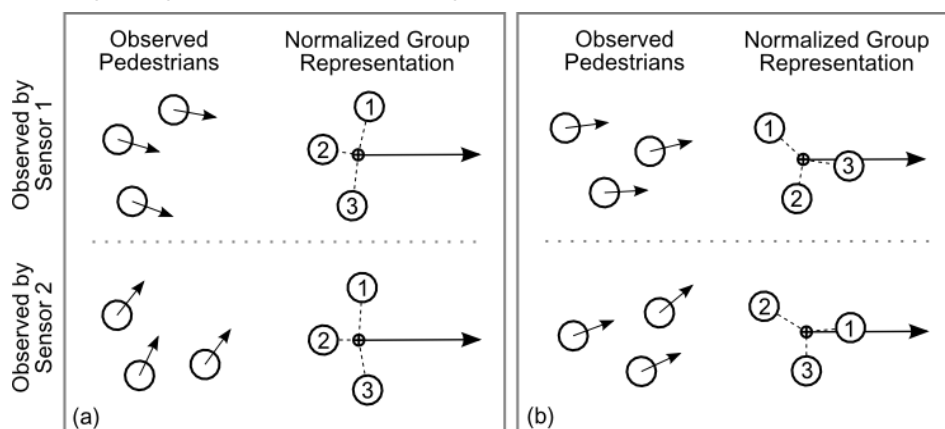


Figure 5. Matching member vectors between two groups. Each diagram shows a group of observed pedestrians with noisy position and direction, and the representation of that group normalized by group center and group direction. (a) unambiguous case. (b) ambiguous case where counterclockwise counting would not correctly compare the groups.

However, in some groups, noisy readings for motion direction may cause the counterclockwise pairing strategy to fail, as illustrated in Fig. 5 (b). We can avoid this by using nearest-neighbor matching to identify the first member for the second group. Once one pair of members is associated, we can iterate through the remaining members without repeating the nearest-neighbor search.

For two groups ( $G1, G2$ ) being compared, define  $\vec{m}_i^{G1}$  as the  $i^{th}$  member of  $G1$  and  $\vec{m}_i^{G2}$  as the corresponding member of  $G2$ . Once correspondences have been established, we compute  $d_i = \|\vec{m}_i^{G1} - \vec{m}_i^{G2}\|$  for each pair. If all pairs satisfy  $d_i < d_{threshold}$  we consider groups  $G1$  and  $G2$  to

be a valid match.

Fig. 6 shows an example of the two groups shown in Fig. 2 (bottom), from the perspective of two different sensors, “L” and “N.” Due to occlusions, two pedestrians are not visible to Sensor N. Table 2 shows the group feature vectors visible to each sensor. The two-person “Group 3” seen by Sensor N can be matched with “Subgroup 2c” of the 3-person “Group 2” observed by Sensor L.

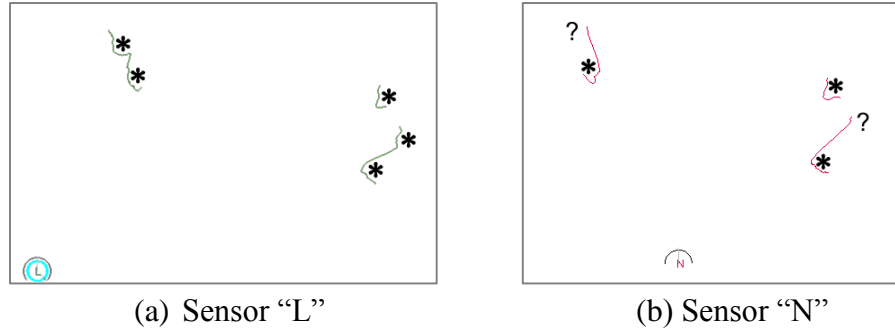


Figure 6. Point-cloud data for two group detections. Visible people are marked with a “\*” and occluded people with a “?”.

**Table 2:** Groups detected by two sensors in example. Corresponding groups are highlighted.

	# Members	$\ \vec{v}\ $ in mm/s	$\vec{m1}$ (r,θ)	$\vec{m2}$ (r,θ)	$\vec{m3}$ (r,θ)
<b>Sensor L</b>					
Group 1	2	940	(341, 107°)	(341, 287°)	
Group 2	3	718	(674, 58°)	(276, 159°)	(677, 262°)
Subgroup 2a	2	710	(388, 37°)	(388, 217°)	
Subgroup 2b	2	696	(393, 102°)	(393, 282°)	
Subgroup 2c	2	789	(661, 70°)	(661, 250°)	
<b>Sensor N</b>					
Group 3	2	768	(659, 72°)	(659, 252°)	

### 3.3.2. Generating a hypothesis

For a given group observation  ${}^{S1}G1$  observed by sensor S1, and a second observation  ${}^{S2}G1$  of that same group, observed by sensor S2, we can define a hypothesis for the sensor offset in the form of a transformation matrix  ${}^{S1}H_{S2}$ . The rotational offset  ${}^{S1}\theta_{S2}$  is equal to the difference of the motion directions  ${}^{S1}\theta_{G1}$  and  ${}^{S2}\theta_{G1}$ , and the translational offset can be found as a difference of the group center points,  $({}^{S1}x_{G1}, {}^{S1}y_{G1})$  and  $({}^{S2}x_{G1}, {}^{S2}y_{G1})$ .

### 3.4. Cluster identification by Hough Transform voting

The next step is to use a Hough Transform to identify clusters of matches that vote for similar relative sensor offsets. To do this, we define a discrete accumulator grid in  $x, y, \theta$ -space, and for each social group match that is identified, we add votes to the bin corresponding to the transformation hypothesis determined by that group.

We can reduce the filter's susceptibility to noise by weighting the number of votes for each hypothesis according to its likelihood of correctness. To estimate this likelihood, we define a consistency metric  $C(x, y, \theta)$  based on the recorded history of observations from the two sensors. For each time slice in the recorded data, consider a set of 2-dimensional points  ${}^{S1}\mathbf{p}_1$  detected by sensor S1, a set of points  ${}^{S2}\mathbf{p}_2$  detected by sensor S2, and a proposed hypothesis for the transformation matrix  ${}^{S1}H_{Gn}$  based on a shared observation of group  $G_n$ . Multiplying  ${}^{S1}H_{Gn} {}^{S2}\mathbf{p}_2$  yields the set of points  ${}^{S1}\mathbf{p}_2$  in the coordinate system of sensor S1, so they can be directly compared with  ${}^{S1}\mathbf{p}_1$ , with which they should overlap if the hypothesis is correct.

To find matching observation pairs, we filter by motion direction of the pedestrian, such that only observation pairs having motion directions within a threshold angle of each other, i.e.

$$\left| {}^{S1}\theta_{p2}^{(i)} - {}^{S1}\theta_{p1}^{(j)} \right| > \theta_{match},$$

are considered as potential matches.

For each remaining point in  ${}^{S1}\mathbf{p}_2$ , a nearest-neighbor search is performed among all points in  ${}^{S1}\mathbf{p}_1$ . For computational efficiency, a k-d tree is used for this search, as suggested in [27]. The distance from a point  ${}^{S1}\mathbf{p}_2^{(i)}$  to its nearest neighbor  ${}^{S1}\mathbf{p}_1^{(j)}$  and second-nearest neighbor  ${}^{S1}\mathbf{p}_1^{(k)}$  are calculated. The nearest neighbor is considered a match if the Euclidean distance ratio is less than or equal to 0.8, that is,  $\frac{\|{}^{S1}\mathbf{p}_2^{(i)} - {}^{S1}\mathbf{p}_1^{(j)}\|}{\|{}^{S1}\mathbf{p}_2^{(i)} - {}^{S1}\mathbf{p}_1^{(k)}\|} \leq 0.8$ . This is based on the technique used by Lowe for keypoint matching, which was reported to eliminate 90% of the false matches while discarding less than 5% of correct matches [24].

The consistency metric  $C(x, y, \theta)$  is then defined as the total number of matches between  ${}^{S1}\mathbf{p}_1$  and  ${}^{S1}\mathbf{p}_2$ , aside from the points contained in group  $G_n$  itself. For hypotheses that are far from the true offset between the sensors,  $C$  will be low or zero, and for hypotheses that are close to the true value,  $C$  will be high. For each hypothesis we add  $C$  votes to the accumulator bin.

Finally, we output the hypothesis corresponding to the highest-scoring accumulator bin (or, if there is not one highest-scoring bin, by taking the union of the sets of pairs associated with all the top bins). Together with this hypothesis, we output the matching observation pairs discovered during the consistency check for that grid element.

### 3.5. Pairwise model verification and outlier rejection

Once enough data has been collected that at least one bin contains data from a threshold number of

distinct groups (we obtained good results by requiring 5), a sensor pair is considered to have a reliable hypothesis. Once every sensor has at least one reliable hypothesis linking it to the network, the RANSAC (random sample consensus) technique [28] is used to refine each hypothesis through outlier rejection.

For each sensor pair, we use the best hypothesis and corresponding set of observation pairs obtained in the Hough transform step. As this hypothesis may be slightly incorrect, the set of observation pairs may contain false matches, as illustrated in Fig. 7. The objective of this step is to reject these false matches by further refining the sensor offset hypothesis and the set of corresponding observation pairs.

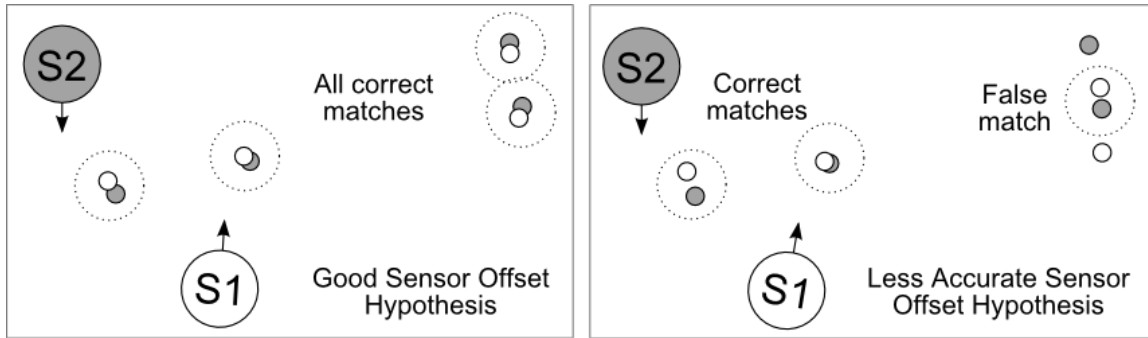


Figure 7. Example of a false match between observations based on nearest-neighbor matching.

The RANSAC technique is an iterative process in which a random subset of observation pairs is chosen, and a best-fit sensor offset hypothesis  $H'$  is fitted to that data set. The remaining data points are classified as inliers or outliers based on whether they have nearest-neighbor matches assuming  $H'$ . A refined sensor offset hypothesis  $H''$  is recomputed using the set of inliers, and the Cartesian error  $\epsilon$  is computed over all observation pairs marked as inliers using that hypothesis. After several iterations, the hypothesis  $H''$  that minimizes  $\epsilon$  is accepted as the best model, and its associated set of inliers is stored for the next step in the algorithm. This process is repeated for all sensor pairs with at least some threshold number of shared observations.

### 3.6. Simultaneous solution of network constraints

The final step is to combine the refined set of inliers for each of the sensor pairs into a constraint matrix, which can be solved for the relative positions of all sensors in the network. To build the constraint matrix, we consider each sensor  $n$ , with absolute position  $(x_n, y_n)$  and orientation  $\theta_n$ . To transform from the coordinate system of sensor  $n$  into global coordinates requires a rotation of  $-\theta_n$  and translations of  $-x_n$  and  $-y_n$ , represented by the homogeneous transformation matrix in Eq. 1.

$${}^0\mathbf{T}_n = \begin{bmatrix} \cos\theta_n & \sin\theta_n & -x_n \\ -\sin\theta_n & \cos\theta_n & -y_n \\ 0 & 0 & 1 \end{bmatrix} \quad (1)$$

The coordinates and orientations of all  $n$  sensors can be represented as a parameter vector  $\boldsymbol{\beta} =$

$[x_1 \ y_1 \ \theta_1 \ x_2 \ y_2 \ \theta_2 \ \dots \ x_n \ y_n \ \theta_n]^T$ . Each human observation in sensor-relative coordinates,  ${}^n p$ , can now be transformed to the global coordinate system:  ${}^0 p = {}^0 T_n {}^n p$

For each sensor pair, we consider pairs of corresponding shared observations  $\{{}^1 p, {}^2 p\}$  which should represent the same points in the global coordinate system. Hence, the error between them should be minimized:

$$\text{Given shared observations } {}^1 p = \begin{bmatrix} {}^1 p_x \\ {}^1 p_y \\ 1 \end{bmatrix}, {}^2 p = \begin{bmatrix} {}^2 p_x \\ {}^2 p_y \\ 1 \end{bmatrix}$$

$$\text{Minimize the error function } \varepsilon(\boldsymbol{\beta}) = \| {}^0 T_1 {}^1 p - {}^0 T_2 {}^2 p \|^2$$

Next, we combine these constraints into a single constraint matrix.

$$\text{For one } 6 \times 1 \text{ shared observation vector } p_{1,2} = [{}^1 p^T \quad {}^2 p^T]^T$$

$$[{}^0 T_1 \quad -{}^0 T_2] p_{1,2} = \begin{bmatrix} {}^0_1 p_x - {}^0_2 p_x \\ {}^0_1 p_y - {}^0_2 p_y \\ 0 \end{bmatrix}$$

$$\text{For } m \text{ points, } \left\| \begin{bmatrix} {}^0 T_1 & -{}^0 T_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & {}^0 T_1 & -{}^0 T_2 \end{bmatrix} [p_{1,2}^1 \quad p_{1,2}^2 \quad \dots \quad p_{1,2}^m]^T \right\| = \varepsilon_{1,2}(\boldsymbol{\beta})$$

Define these elements as the  $3m \times 6$  difference matrix:

$$\mathbf{D}_{1,2} = \begin{bmatrix} {}^0 T_1 & -{}^0 T_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & {}^0 T_1 & -{}^0 T_2 \end{bmatrix}$$

$$\text{And the shared } 6 \times n \text{ observation matrix } \mathbf{p}_{1,2} = [p_{1,2}^1 \quad p_{1,2}^2 \quad \dots \quad p_{1,2}^n]^T$$

Now, combine observations from other sensor pairs:

$$\left\| \begin{bmatrix} \mathbf{D}_{1,2} & & 0 \\ & \mathbf{D}_{1,3} & \\ & & \ddots \\ 0 & & & \mathbf{D}_{n-1,n} \end{bmatrix} \begin{bmatrix} \mathbf{p}_{1,2} \\ \mathbf{p}_{1,3} \\ \vdots \\ \mathbf{p}_{n-1,n} \end{bmatrix} \right\| = \varepsilon(\boldsymbol{\beta}) \quad (2)$$

Then we use the Levenberg–Marquardt algorithm to find the optimal values of  $\boldsymbol{\beta}$  that minimize  $\varepsilon(\boldsymbol{\beta})$  in Eq. 2 via least-squares minimization.

## 4. EVALUATION

To evaluate the performance of the algorithm, we measured calibration accuracy of sensor positions compared with ground truth in a controlled environment. We then measured tracking

accuracy attained by using the proposed technique in three public locations. Both analyses were performed offline, using recorded raw sensor data.

#### 4.1. Sensor Position Accuracy

To provide a reference for ground truth, we set up four sensors in precisely-measured positions in a section of hallway in our laboratory, as shown in Fig. 8.

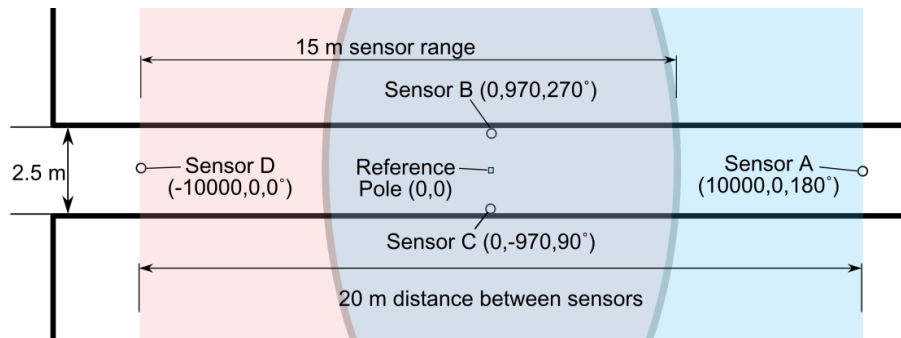


Figure 8. Sensor placement for position accuracy measurement trials.

The sensors were placed as close as possible to the nominal positions. To align sensor angles, a reference pole was placed at the origin, and sensors were manually rotated until the reference pole coincided with the center scan point detected by the laser range finders. Exact offsets of the sensors were then measured using a Bosch DLE 150 precision laser range measurement device, and sensor angles were fine-tuned in software to align the reference pole and wall detections.

Five trials were conducted in which five people, members of our laboratory, walked through the corridor ten times each in groups of two and three. Data from each of the sensors was then replayed offline and calibrated using our system.

The results, presented in Table 3, show an average displacement error of 34 mm and an average angular error of 0.51 degrees, averaged over the four sensors. We were quite satisfied with this result. However, these results do not directly answer the question of whether this level of accuracy is sufficient for precise tracking of pedestrians.

#### 4.2. Evaluation of Tracking Accuracy

While accuracy of sensor position estimates is an important evaluation of our calibration technique, its ultimate purpose is to enable consistent estimates of pedestrian positions from multiple sensors. To evaluate the level of tracking accuracy made possible by our technique, we performed three tests of our system in public spaces.

**Table 3.** Sensor Position Accuracy Results

Trial	Displacement Error (mm)	Angular Error (degrees)
-------	-------------------------	-------------------------

1	38	0.70
2	26	0.86
3	61	0.32
4	20	0.26
5	20	0.43
<b>Average</b>	<b>34</b>	<b>0.51</b>

For each location, we used separate data sets for calibration and evaluation. Absolute ground truth of the pedestrians' positions was not available, so we based our consistency evaluation on the centroid of estimates from different sensors. Let  $p_s^{(i)}(t)$  represent the estimated position of person  $i$  at time  $t$ , as observed by sensor  $s$ , with centroid  $\hat{p}^{(i)}(t)$  computed among all observations from different sensors. The number of sensors  $S^{(i)}(t)$  which can observe a person  $i$  at any given position depends on geometry and dynamic occlusions. We compute the average error  $\varepsilon$  from the centroid, in Eq. 3.

$$\varepsilon^{(i)}(t) = \frac{\sum_s \left\| \hat{p}^{(i)}(t) - p_s^{(i)}(t) \right\|}{S^{(i)}(t)} \quad (3)$$

A set of reference pedestrians was obtained by manually identifying corresponding trajectories from different sensors. 150 reference pedestrians were identified for each environment. For each reference pedestrian, we evaluated  $\varepsilon^{(i)}(t)$  at each time step during which the pedestrian was simultaneously observed by at least two sensors.

To visualize the consistency of tracking as a function of spatial location, we divided the space into a grid of 1m resolution. Each grid element was assigned the average of  $\varepsilon^{(i)}(t)$  across all data points for which the centroid fell within that grid element, and the results are shown below for each environment. To obtain a final metric of accuracy, we computed the average error over all grid elements for each environment.

For comparison, we have also included the accuracy results for our best-effort manual calibration based on visual inspection of scan data, which is typical of the calibration accuracy available to us before developing this technique.

### 4.3. Results

#### 4.3.1. Straight corridor

The first environment we analyzed (“*Diamor*”) was a corridor approximately 50m long and 7m wide in *Diamor* Osaka, an underground shopping area located between several train and subway stations in Osaka, shown in Fig. 9. This environment was geometrically simple, with a high degree of overlap between the coverage regions of 16 sensors lined up along a straight corridor. 27.6 pedestrians were detected entering this environment per minute. Figure 10 shows the calibration accuracy results, with an average error over all grid elements of 10.4 cm.

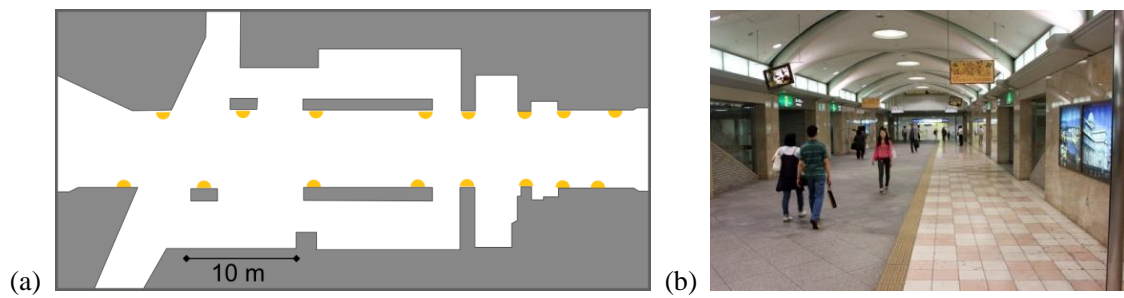


Figure 9. (a) Map of the *Diamor* environment. (b) Photo of the *Diamor* environment.

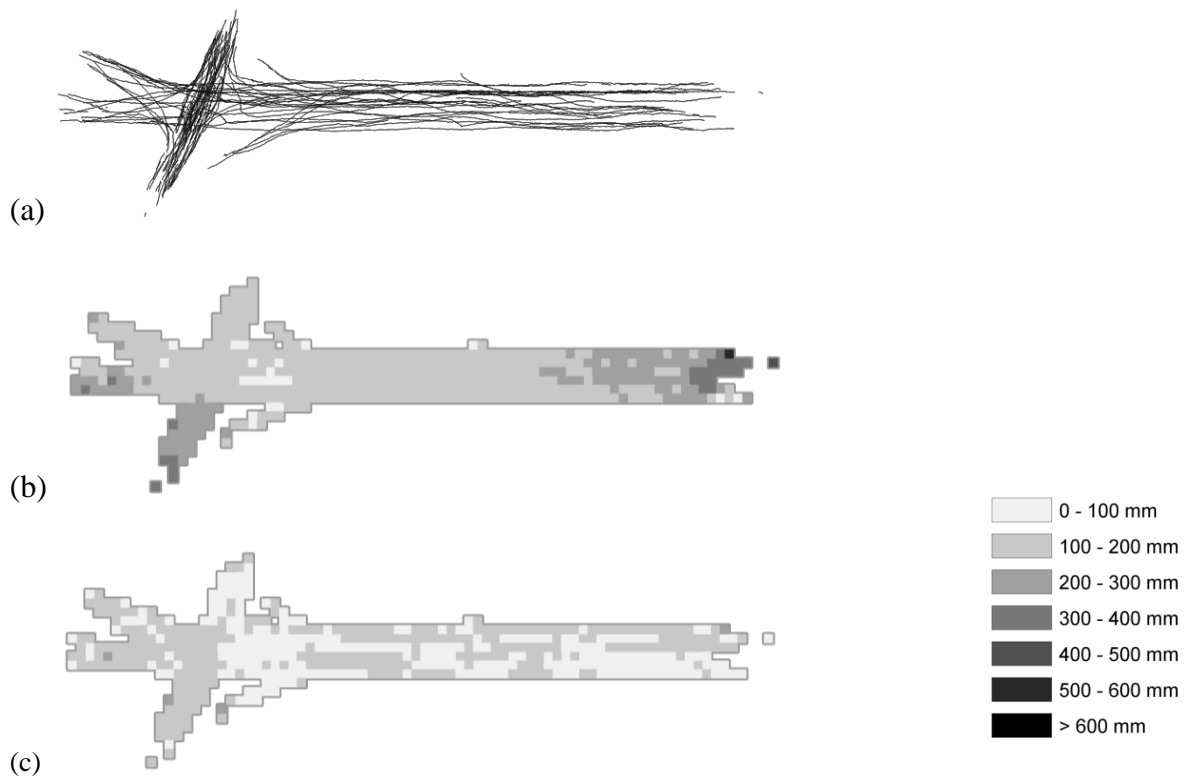


Figure 10. Calibration results for the *Diamor* environment. (a) Trajectories used in the accuracy analysis. (b) Spatial error from manual calibration. (c) Spatial error from automatic calibration.

#### 4.3.2. Large space with multiple entrances

The second environment (“ATC”) was a space over 60 m long and 25 m wide at its widest point, consisting of a hallway opening into large atrium at the Asia and Pacific Trade Center, a shopping and wholesale trade complex on the Osaka waterfront, shown in Fig. 11. An average of 31.8 pedestrians entered the environment per minute. The sensor network consisted of 19 sensors. Figure 12 shows the calibration accuracy results, with an average error over all grid elements of 8.3 cm.



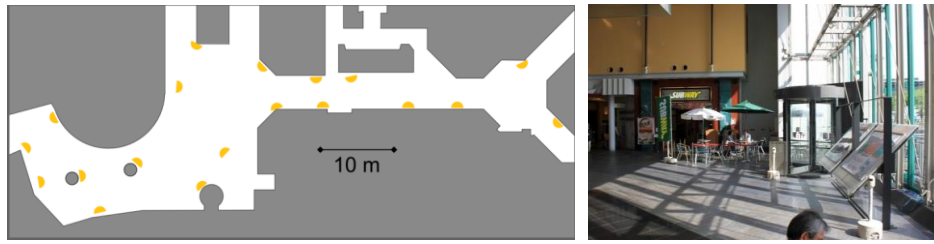


Figure 11. (left) Map of the ATC environment. (right) Photo of atrium area.

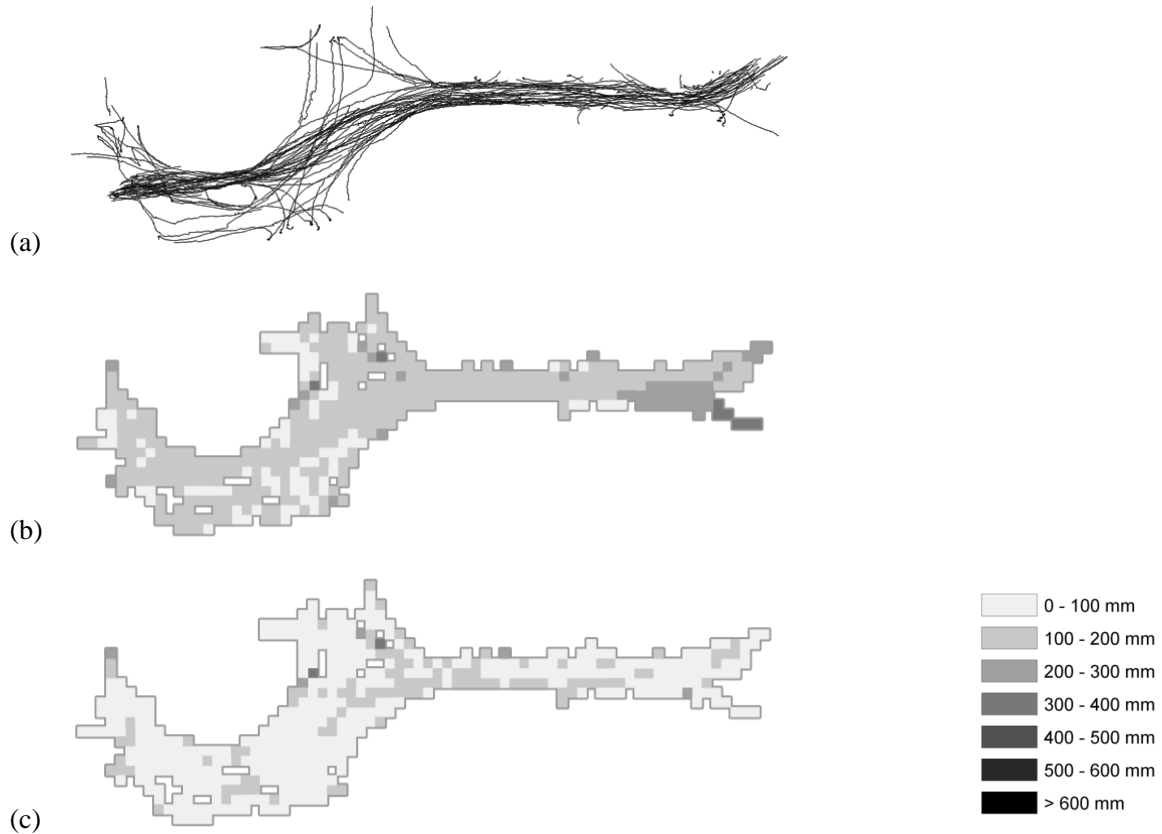


Figure 12. Calibration results for the *ATC* environment. (a) Trajectories used in the accuracy analysis. (b) Spatial error from manual calibration. (c) Spatial error from automatic calibration.

#### 4.3.3. Subdivided space with many occlusions

The third environment we analyzed (“*Apita*”) was a 15m by 10m space inside the entrance to the APiTA Town Keihanna Shopping Center, a shopping mall near our laboratory, shown in Fig. 13. This area was only observed by 8 sensors.

Although this environment was physically smaller than the previous two environments, it was more difficult to calibrate, due to obstacles in the environment such as shopping carts, sliding doors, and clothing racks. There were also fewer pedestrians than the other two environments, with 22.7 people entering the space each minute. Figure 14 shows the calibration accuracy results, with an average error over all grid elements of 7.9 cm.



Figure 13. (a) Map of the *Apita* environment. (b) Photo of the *Apita* environment.

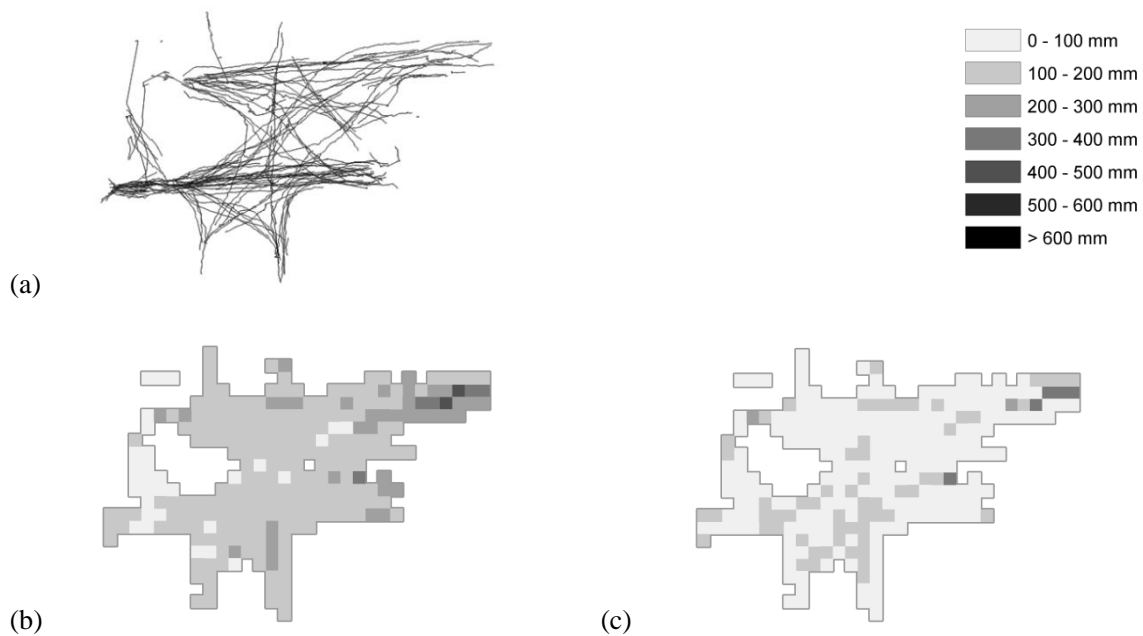


Figure 14. Calibration results for the *Apita* environment. (a) Trajectories used in the accuracy analysis. (b) Spatial error from manual calibration. (c) Spatial error from automatic calibration.

#### 4.4. Summary of results

The graph in Fig. 15 shows the average calibration accuracy in each of the three environments. The proposed technique yielded results superior to the manual calibration estimates which had actually been used for each dataset, resulting in tracking accuracy within 11 cm in every case.

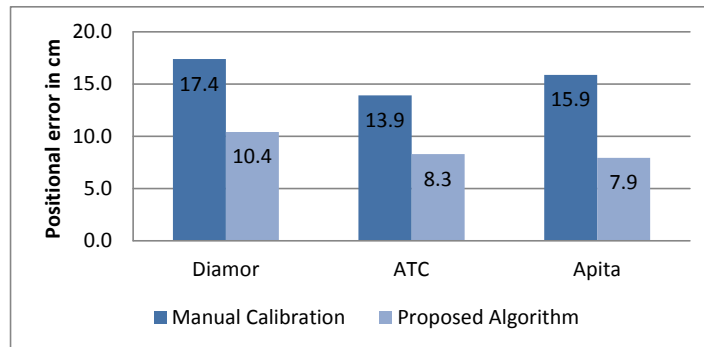


Figure 15. Overall calibration accuracy compared with manual calibration for each environment.

#### 4.5. Analysis of steps in the algorithm

To illustrate the relative effectiveness of each of the steps in our algorithm, Fig. 16 shows the alignment of human positions detected by two sensors at one time frame, based on sensor offset hypotheses output by the Hough transform step, RANSAC step, and final calibration.

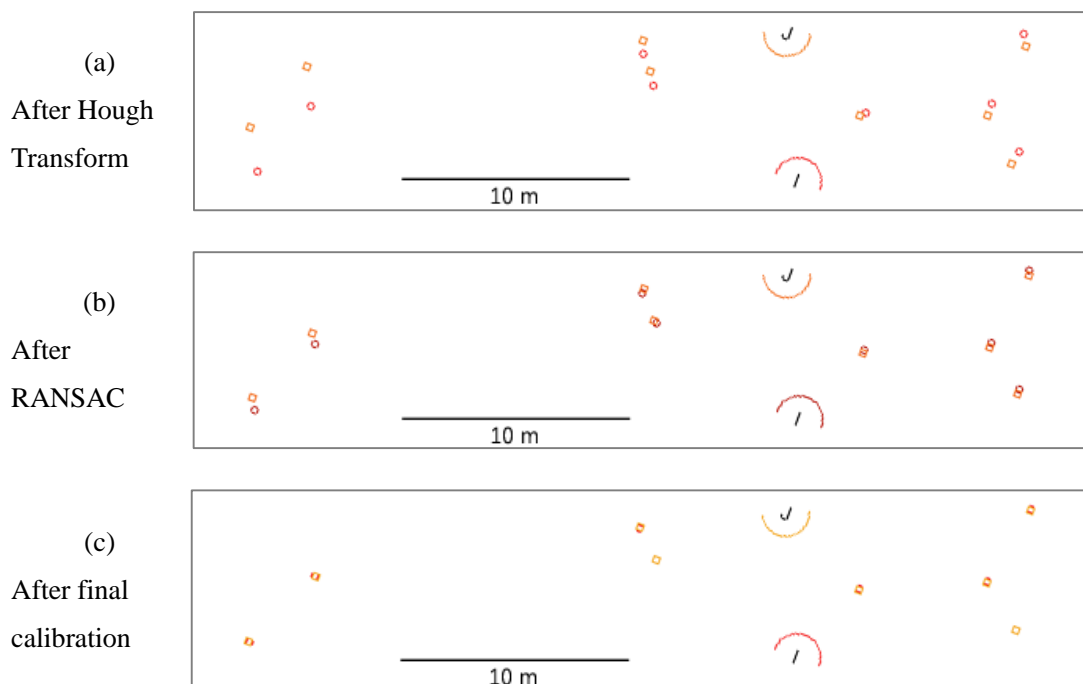


Figure 16. Alignment of simultaneous human detections by two sensors (a) based on Hough transform output, (b) based on RANSAC output, (c) based on final calibrated positions in a 16-sensor network.

Recall that in our algorithm, each sensor offset hypothesis is used to generate a set of proposed matches between human observations, which are eventually used to generate the constraint matrix. The accuracy of final calibration strongly depends on the elimination of false matches in this matrix. To illustrate how each step in the algorithm reduces false matches, we computed confusion matrices

for three stages of the calibration process: group matching, the Hough transform, and RANSAC, based on a second set of ground truth trajectories of reference pedestrians, taken from the data set used to perform calibration. The results are shown in Table 4. As these results show, each step substantially reduces the number of false positives, contributing to the accuracy of the final calibration.

**Table 4.** Percentage of correct human observation matches generated at each step of the algorithm, evaluated for each of the three environments.

	Diamor		ATC		Apita	
	Correct	Incorrect	Correct	Incorrect	Correct	Incorrect
Group Matching	78.3 %	21.7 %	66.8 %	33.3 %	71.6 %	28.4 %
Hough transform	97.0 %	3.0 %	94.3 %	5.7 %	91.9 %	8.1 %
RANSAC	99.6 %	0.4 %	99.3 %	0.8 %	99.1 %	0.9 %

## 5. DISCUSSION

### 5.1. Areas of low pedestrian traffic

This algorithm was designed for locations with high pedestrian traffic – one motivation for using social groups as keypoint features was to avoid the computational load required to compare many individual pedestrians. However, in low-traffic areas, computation would not be a significant problem, so it could make sense to use all individual trajectories as keypoint features.

### 5.2. Multiple iterations

The proposed procedure produces a sparse matrix of sensor-pair hypotheses which can be solved relatively quickly. A two-pass solution could also be considered, where the sensor-pair hypotheses estimated in the first pass are fed back into the RANSAC step of a second pass. This technique produces a dense matrix which should yield higher calibration accuracy, but which will take much more CPU time to solve.

### 5.3. Pitch and roll

This technique could be extended to take into account pitch and roll angles of the sensors, by modeling the world as vertically invariant and projecting scan points into a horizontal plane. Because our sensors are rigidly mounted on sturdy metal bases and used on level floors, such compensation was unnecessary in our work. It is likely that some error would be introduced by modeling pedestrians as vertically invariant.

### 5.4. Sensor placement

The key to fast and accurate calibration is generating a large number of shared observations, e.g.

when most space is covered by more than two sensors, such as in the Diamor environment. “Weak links” like the sliding doors separating the three left sensors from the rest in the Apita space make calibration difficult. If only one door existed, it is likely that the small room to the left would be internally consistent but globally misaligned, but the fact that two sliding doors exist helps to solve the problem by enabling loop closure.

## 6. CONCLUSIONS

This study has demonstrated a technique for calibrating the positions of laser range finders for a pedestrian-tracking system which can be used for studying crowd dynamics as well as assisting robots in navigational interactions with humans. This technique improves on a previously proposed pedestrian-based calibration technique by considering social groups, providing a smaller search space and more features for inter-sensor observation matching than using pedestrians alone.

We have shown that this technique can calibrate large networks of up to 19 laser range finders, yielding an average tracking accuracy within 11 cm of error. These results validate the algorithm proposed in this paper and illustrate an example of how features of social pedestrian dynamics can be considered in a similar way to physical features in the environment.

## ACKNOWLEDGMENTS

We would like to thank Yoshifumi Nakagawa, Tetsushi Ikeda, and Satoshi Koizumi for their work in organizing the collection of the data used in this study. We would also like to thank the staff at ATC, Diamor Osaka, and APiTA Town Keihanna for their support and cooperation. This research was supported by the Ministry of Internal Affairs and Communications of Japan with a contract entitled “Novel and innovative R&D making use of brain structures.”

## REFERENCES

- [1] Fod, A., A. Howard, and M.A.J. Mataric, A laser-based people tracker. in *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, pp. 3024-3029 (2002)
- [2] Panangadan, A., M. Mataric, and G. Sukhatme, Detecting anomalous human interactions using laser range-finders. in *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, pp. 2136-2141 vol.3 (2004)
- [3] Zhao, H. and R. Shibasaki, A novel system for tracking pedestrians using multiple single-row laser-range scanners. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on* **35**(2), p. 283-291 (2005)
- [4] Cui, J., et al., Laser-based detection and tracking of multiple people in crowds.

- Comput. Vis. Image Underst.* **106**(2-3), p. 300-312 (2007)
- [5] Prassler, E., J. Scholz, and P. Fiorini, Navigating a Robotic Wheelchair in a Railway Station during Rush Hour. *The International Journal of Robotics Research* **18**(7), p. 711-727 (1999)
- [6] Montemerlo, M., S. Thrun, and W. Whittaker, Conditional particle filters for simultaneous mobile robot localization and people-tracking. in *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, pp. 695-701 vol.1 (2002)
- [7] Schulz, D., et al., People Tracking with Mobile Robots Using Sample-Based Joint Probabilistic Data Association Filters. *The International Journal of Robotics Research* **22**(2), p. 99-116 (2003)
- [8] Arras, K.O., O.M. Mozos, and W. Burgard, Using boosted features for the detection of people in 2d range data. in *Robotics and Automation, 2007 IEEE International Conference on*, pp. 3402-3407 (2007)
- [9] Xavier, J., et al., Fast line, arc/circle and leg detection from laser scan data in a player driver. in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pp. 3930-3935 (2005)
- [10] Kanda, T., et al., Who will be the customer?: a social robot that anticipates people's behavior from their trajectories. in *Proceedings of the 10th international conference on Ubiquitous computing*, Seoul, Korea, pp. 380-389 (2008)
- [11] Glas, D.F., et al., Simultaneous people tracking and localization for social robots using external laser range finders. in *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*, St. Louis, MO, USA, pp. 846-853 (2009)
- [12] Ikeda, T., et al., Person identification by integrating wearable sensors and tracking results from environmental sensors. in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pp. 2637-2642 (2010)
- [13] Satake, S., et al., How to approach humans?: Strategies for social robots to initiate interaction. in *Proceedings of the 4th ACM/IEEE international conference on Human robot interaction*, La Jolla, California, USA, pp. 109-116 (2009)
- [14] Senior, A.W., A. Hampapur, and M. Lu, Acquiring multi-scale images by pan-tilt-zoom control and automatic multi-camera calibration. in *Application of Computer Vision, 2005. WACV/MOTIONS'05 Volume 1. Seventh IEEE Workshops on*, pp. 433-438 (2005)
- [15] Hightower, J. and G. Borriello, Location systems for ubiquitous computing. *Computer* **34**(8), p. 57-66 (2001)
- [16] Dellaert, F., et al., Monte Carlo localization for mobile robots. in *Robotics and*

- Automation, 1999. Proceedings. 1999 IEEE International Conference on*, pp. 1322-1328 vol.2 (1999)
- [17] Durrant-Whyte, H. and T. Bailey, Simultaneous localization and mapping: part I. *Robotics & Automation Magazine, IEEE* **13**(2), p. 99-110 (2006)
- [18] Glas, D.F., et al., Automatic position calibration and sensor displacement detection for networks of laser range finders for human tracking. in *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*, pp. 2938-2945 (2010)
- [19] Sogo, T., *Localization of Sensors and Objects in Distributed Omnidirectional Vision*, in *Department of Social Informatics 2001*, Kyoto University. p. 155.
- [20] Henderson, L.F., The Statistics of Crowd Fluids. *Nature* **229**(5284), p. 381-383 (1971)
- [21] Helbing, D. and P. Molnár, Social force model for pedestrian dynamics. *Physical Review E* **51**(5), p. 4282-4286 (1995)
- [22] Zanlungo, F., T. Ikeda, and T. Kanda, Social force model with explicit collision prediction. *EPL (Europhysics Letters)* **93**(6), p. 68005 (2011)
- [23] Moussaïd, M., et al., The Walking Behaviour of Pedestrian Social Groups and Its Impact on Crowd Dynamics. *PLoS ONE* **5**(4), p. e10047 (2010)
- [24] Lowe, D.G., Object recognition from local scale-invariant features. in *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, pp. 1150-1157 vol.2 (1999)
- [25] Costa, M., Interpersonal Distances in Group Walking. *Journal of Nonverbal Behavior* **34**(1), p. 15-26 (2010)
- [26] Yücel, Z., et al., Deciphering the Crowd: Modeling and Identification of Pedestrian Group Motion. *Sensors* **13**(1), p. 875-897 (2013)
- [27] Besl, P.J. and H.D. McKay, A method for registration of 3-D shapes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **14**(2), p. 239-256 (1992)
- [28] Fischler, M.A. and R.C. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* **24**(6), p. 381-395 (1981)