The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

Automatic Position Calibration and Sensor Displacement Detection for Networks of Laser Range Finders for Human Tracking

Dylan F. Glas, Takahiro Miyashita, Hiroshi Ishiguro, *Member, IEEE*, and Norihiro Hagita, *Senior Member, IEEE*

***Abstract*—Laser range finders are a non-invasive tool which can be used for anonymously tracking the motion of people and robots in real-world environments with high accuracy. Based on a commercial system we have developed, this paper addresses two practical issues of using networks of portable laser range finders in field environments. We first describe a technique for automated calibration of sensor positions and orientations, by using velocity-based matching of observed human trajectories to define constraints between the sensors. We then propose a mechanism for detecting when a sensor has been moved out of alignment, which can be used to alert an operator of the condition and automatically exclude erroneous data from tracking calculations.**

After describing our techniques for solving these problems, we demonstrate the effectiveness of our calibration and error detection systems in live trials with our real-time system, as well as offline tests based on scan data recorded from field trials.

I. Introduction

Laser range finders (LRF's) have become an indispensable tool in the robotics community, the core component of many localization, mapping, and obstacle avoidance systems. A less common but growing application field for laser range finders is the tracking of human motion, particularly through public or commercial spaces.

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. Other advantages include high measurement accuracy and simplicity of data processing when compared with visual tracking. Additionally, the fact that LRF-based trackers output only anonymous range values enables analysis of the social use of spaces while putting security advocates at ease to some degree. While these benefits must be balanced against the cost of the sensors, they remain a promising tool for analyzing human motion in high-traffic public spaces.

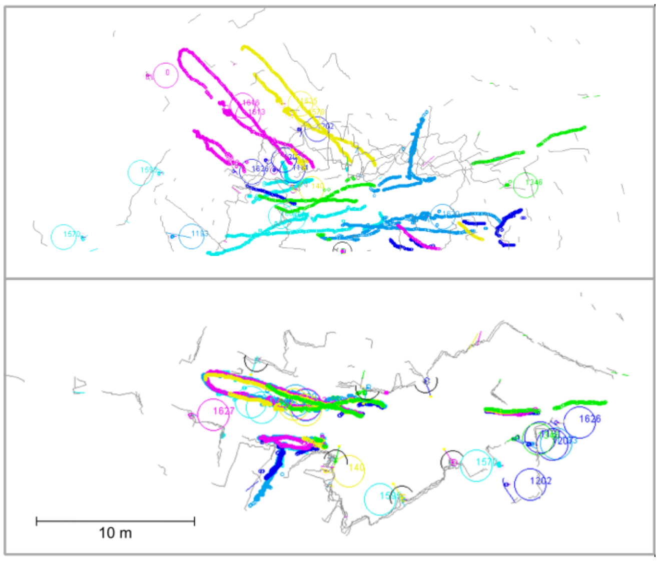

Fig. 1. Solid colored lines show human trajectories seen by each sensor before calibration (top) and after calibration (bottom) using 30 seconds of trajectory data. Background scans are shown in gray, for reference only; no scan matching was performed.

These sensors have been used to track people in order to study human motion [1]-[4] and to enable autonomous robot operation in populated environments [5]-[7].

Our laboratory has developed a human tracking system based on a stationary network of portable laser range finders. This system is now a commercial product, and it has been used by a variety of research labs and corporations in market studies and field trials. It has been used to analyze the use of social spaces, to make predictions about human motion [8], to support robot localization [9], to locate specific individuals [10], and to enable robots to approach or avoid people [11].

To provide reliable data for these tasks, precise knowledge of the positions and orientations of the sensors is critical. In this paper we present techniques for automatic calibration of sensor positions (Fig. 1) and fast detection of sensors which have been moved out of alignment.

The techniques presented in this paper have been tested with our own system. Other tracking systems may differ in various ways, such as the use of leg-height sensors as opposed to the torso-height sensors we use, but the solutions proposed in this paper should be generalizable to any tracking system which depends on scan registration between multiple range-based sensors.

Manuscript received March 11, 2010. This work was supported by KAKENHI (21118003).

D. F. Glas, T. Miyashita, and N. Hagita are with ATR Intelligent Robotics and Communication Labs., Kyoto, Japan (corresponding author's phone: +81-774-95-1424; fax: +81-774-95-1408; e-mail: dylan@atr.jp).

H. Ishiguro is with the Intelligent Robotics Laboratory at the Graduate School of Engineering Science at Osaka University, Osaka 565-0871, Japan, and the Intelligent Robotics and Communication Laboratories at ATR. (e-mail: ishiguro@sys.es.osaka-u.ac.jp).

978-1-4244-6676-4/10/$25.00 ©2010 IEEE

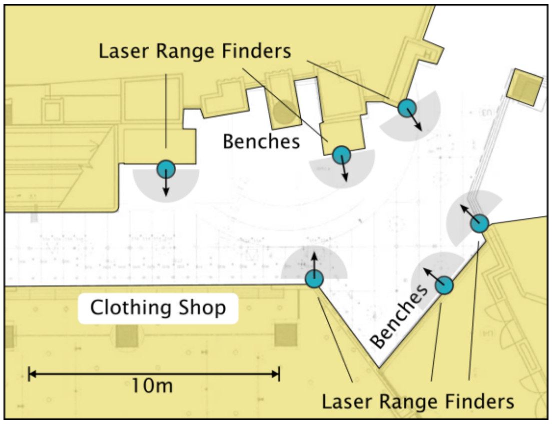

Fig. 2. Operational environment for our field trials. Six laser range finders monitored the central area, where robots approached customers and offered directions and recommendations.

II. Background and Requirements

The calibration and error detection problems addressed in this paper are motivated by our field experiences with a human tracking system in the field. To provide context for this discussion, we will briefly describe the system used in our field trials at Universal CityWalk Osaka, a shopping area outside of Universal Studios Japan.

A. System Overview

In the environment shown in Fig. 2, we placed six SICK LMS-200 laser range finders around the perimeter of an area roughly 20 m long by 5 m wide. They were set to a detection range of 80 m with a nominal precision of 1cm, each scanning an angular area of 180° at a resolution of 0.5°, providing readings of 361 data points every 26 ms.

The laser range finders were mounted 85 cm from the ground, to see over obstacles such as benches and luggage. This height was also chosen to enable robust detection of people at long distances, when scan beams are spaced too far apart to reliably detect legs.

The sensors were housed in heavy steel cases on vertical stands, as shown in Fig. 3. This rigidly fixed the height and tilt of the sensors, but the stands themselves were movable. The sensor positions were thus constrained to a 2D plane parallel to the floor.

All LRF's were connected to a single Linux PC for data acquisition. Raw data was forwarded over a wired LAN to a quad-core Windows XP machine with 4GB of RAM for tracking computations.

The algorithm we use for detecting and tracking humans is described in [12]. In this algorithm, we use simple background subtraction and segmentation (described in Sec. IV-A and IV-B) to identify human-sized detections and initialize particle filters at those points.

Individual particle filters are used to track each person. The likelihood model used by the particle filters integrates edge information and occlusion information (whether or not a point is observed to be empty) from all LRF's.

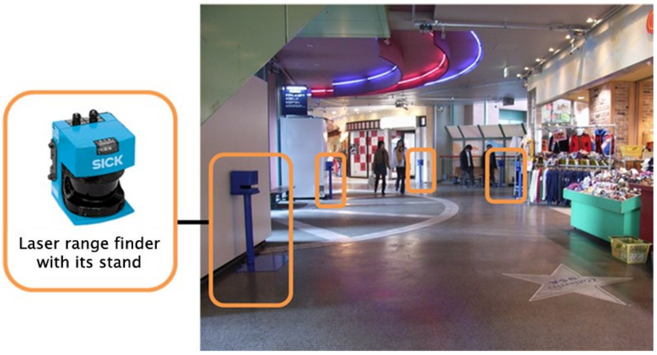

Fig. 3. Mounting stands for laser range finders in the field environment.

B. Calibration Requirements

In order to track human positions, our system combines range data from several independent laser range finders. For accurate registration of the scan data, it is necessary to have good estimates of the relative positions and orientations of the sensors. Even small errors in these estimates can easily lead to inaccurate tracking and systemic false and missed detections.

Our sensors were deployed every morning and retrieved every night. Thus, their positions varied slightly from day to day, and recalibration was necessary each morning. Sensor calibration was a slow, manual process which sometimes required placing objects in the environment for reference. Calibration accuracy was highly dependent upon people's individual abilities, and it was difficult to do in the middle of the day with many people passing through the tracking area.

To address this problem, the first technique we present in this paper is a method for performing this calibration process in an automated, consistent way without disrupting the natural flow of pedestrian traffic.

For our typical deployments with sensors spaced 5-10 meters apart, the system seemed to perform well when the lateral position errors of the sensors were within 10 cm, and when the angular errors were within around 1°. These are only approximate figures, but they provide an order-of-magnitude guideline for calibration accuracy requirements.

C. Error Detection Requirements

In our field trial, the laser range finders were often moved out of alignment by children playing with the sensors, shopkeepers rearranging store displays, and delivery carts being pushed into the sensor stands.

In such cases, the sensor continues to send tracking data that is misaligned with the other sensors as well as its own background scan, causing positioning errors and false human detections.

We were often alerted to these situations only after our robots began showing erratic behavior, such as approaching nonexistent people. By the time the error had been noticed, identified, and corrected, the system had been generating unreliable data for several minutes. Such errors should be detected immediately, to protect the integrity of the tracking calculations.

We propose a technique in this paper which can detect a sensor misalignment within the time span of a single scan frame, immediately remove that sensor from the tracking calculations, and alert an operator about the situation.

III. Related Work

There are many related techniques related to sensor position estimation in existing literature. A survey and taxonomy of techniques and available technologies for sensor localization is provided in [13]. Regarding LRF position calibration in particular, there is a substantial body of related work regarding SLAM and robot localization.

A. Scan-matching techniques

One of the most common techniques by which LRF's are localized is by scan-matching. A number of scan-matching techniques exist, including Iterative Closest Point (ICP) [14], Polar Scan Matching (PSM) [15], and other techniques [16], [17], [18]. In particular, the work of Biber and Straßer [19] regarding *n*-scan matching appears closely related to our problem, as we need to register several scans from different angles. However, their technique still requires an initial estimate of the relative sensor positions, *e.g.* using a mobile robot's odometry.

Though these techniques are related to our problem, there are a number of reasons why we chose not to implement a scan-matching approach.

1) Irresolvable ambiguities

First, ambiguities and symmetries in the environment may lead to false matches between scans. Techniques such as Monte-Carlo Localization [20] and SLAM often address this problem by resolving ambiguities as the robot moves to new locations where unique features are visible. As our system uses stationary sensors, resolving ambiguities by moving the sensor is not possible.

2) Lack of scan features in open spaces

Our sensors often operate in spaces larger than the LRF's maximum detection range. In this case there are no walls or features that can be used for scan matching, and the only shared information will be the scan data of people walking through the open space.

3) Minimal scan overlap between sensors

A third problem is that our sensors are located around the perimeter of the space to be monitored, so sensors are often facing towards each other. Even when their coverage areas overlap, there may be little or no overlap in the features observed in their scan data. For example, when a person walks through the space, the sensors will see opposite sides of the person, so direct scan matching is not possible.

4) Non-level scan planes

Floors and sensor stands are often not perfectly level. If the scan planes of the LRF's are not level, distant objects such as walls tend to appear further from the sensor than they really are. This effect increases over distance, so sensors should be calibrated using data at the average distance of the people to be tracked.

B. Landmark-based techniques

Another family of techniques often used for sensor localization utilizes natural or artificial landmarks in the environment to triangulate the positions of sensors.

A thorough analysis of multi-sensor localization problems and proposes techniques for qualitative localization which could be applied to LRF's is presented in [21], but the main focus is on localization and tracking with omnidirectional cameras.

Other research has focused on localizing sensor networks based on distances and connectivity between the nodes, where the sensors themselves are landmarks [22].

We often performed manual landmark-based localization by eye in our field trials, placing poles in the environment in the mornings before customers arrive. However, the technique presented here uses pedestrians in the environment as landmarks, as it is less invasive to the social environment under observation. In related work, pedestrians have been used for calibration of camera systems in [23], [24].

By correlating the velocity histories of each observed person, we are able to disambiguate human observations and associate them between sensors with no *a priori* knowledge of the relative sensor positions.

C. Error Detection

The authors are not aware of any other attempt to address this particular error detection problem in the literature. This is unsurprising, as it is not an issue for laser range finders which are permanently fixed in place, nor is it a significant problem for movable sensors used for short-term trials or in controlled environments. However, due to the convenience of portable sensor networks, we anticipate an increase in the number of sensor networks like ours as LRF's become more affordable.

IV. Sensor Calibration

Our sensor calibration technique consists of six steps as follows: We first build a background scan for each sensor (Sec. IV-A) and use a segmentation technique to identify likely pedestrian observations in each sensor's scan data (Sec. IV-B). We then link the human observations into trajectories as the pedestrians move over time (Sec. IV-C), and we use their velocity histories to identify when two sensors are observing the same person (Sec. IV-D). When we have identified joint observations between two sensors, we use them to triangulate geometrical constraints between the sensor positions (Sec. IV-E). Finally, we resolve these constraints across the sensor network to determine the relative positions of all sensors (Sec. IV-F).

A. Build a background scan for each sensor

A simple background subtraction and segmenting method is used to identify human positions, although machine learning techniques could possibly be applied as well [25].

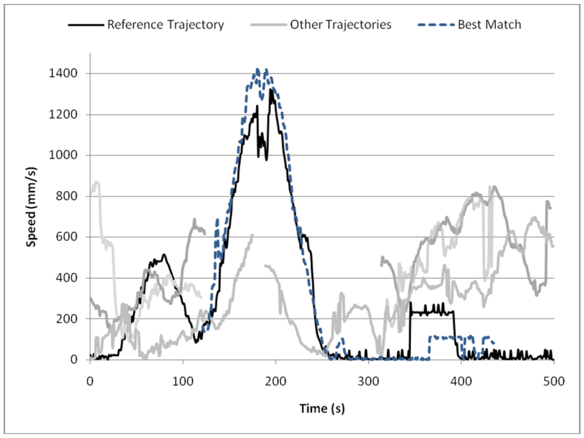

Fig. 4. Velocity profiles of detected trajectories over time. A reference trajectory (black) from one sensor is compared with trajectories from a second sensor. The dashed line shows the best-match trajectory from the second sensor.

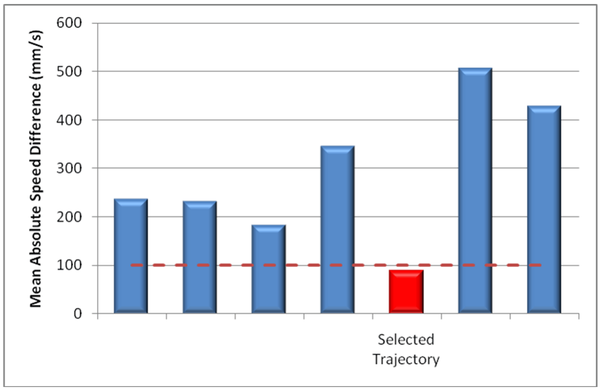

Fig. 5: Average absolute speed difference computed between the reference trajectory and all other trajectories shown in Fig. 4. Comparison trajectories are shown along the x-axis.

A background scan is first 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.

B. Extract human positions from scan data

Each new data scan is then segmented to extract human positions. A median filter is used to remove outliers and smooth the scan data, and then the background scan is subtracted from the data scan, leaving only foreground points.

Continuous segments of foreground points between 30 cm and 80 cm in width are then extracted. Depth discontinuities of more than 30 cm are considered as segment boundaries. Partial occlusions are also considered by artificially extending short segments which end behind observed segment.

Using an elliptical shape model (major axis=55cm, minor axis=30cm), an estimate of the body center position is determined based on the visible body width. This step outputs a list of these human positions in polar coordinates, in the sensor's local coordinate system.

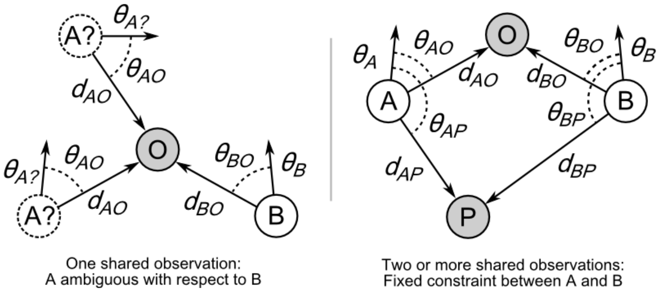

Fig. 6. A single shared observation (O) does not uniquely define the relative locations of two sensors. Adding a second shared observation (P) can eliminate this ambiguity.

C. Generate trajectories for each sensor

At each time step, each of the new human positions is associated, if possible, with a previously observed position.

For each human position in the new dataset, distances are computed to each observation in the previous time step. If a previous observation is the closest match to a new observation and within a threshold distance (50 cm) of that observation, we consider them to be a part of the same trajectory.

D. Associate trajectories between sensors

In order to associate the trajectories between sensors, we compute their velocities over time. To filter the effect of sensor noise and shape irregularities, net velocity is calculated for each trajectory over intervals of 50 time steps.

Next, for each pair of sensors we iterate through all detected trajectories. Each trajectory detected by the first sensor is taken in turn as a "reference trajectory" and compared with every trajectory detected during that time step by the second sensor.

To identify associations between trajectories, the mean error between their velocities is calculated over an interval of 500 time steps. If the mean error exceeds a threshold (we used 100 mm/s), the possible association is discarded. Trajectories with velocities near zero are also removed from the list.

Of the remaining trajectories, if a unique match is found between trajectories observed by two sensors, the association between those trajectories is recorded in a table.

Fig. 4 shows the velocity history for a reference trajectory detected by one sensor, compared with a set of seven trajectories detected by a second sensor for the same time period. Fig. 5 shows the mean speed difference between the reference trajectory and the comparison trajectories over a time window of 500 time steps. Only one trajectory from the second sensor is within a threshold speed difference of 100 mm/s from the reference trajectory, so it is selected as a match.

E. Triangulate constraints for sensor positions

Every 20 time steps, shared observations are recorded from the current positions of each set of associated trajectory pairs. For two sensors A and B with orientation angles θ_A and θ_B, a shared observation O consists of distances d_{AO} and d_{BO} and relative angles θ_{AO} and θ_{BO}. A table of all shared observations is stored for each sensor pair.

Constraints between pairs of sensors are then triangulated from the shared observations tables. As Fig. 6 shows, a single observation is not sufficient for fixing the relative sensor positions. Since only the sensor-relative distance and angle are known, at least two shared observations are required. For a pair of sensors (i, j), a constraint containing the distance d_{ij} and relative directions θ_{ij} and θ_{ji} between them can be computed from two shared observations.

To generate constraints using new data, every new shared observation is paired with a second observation randomly sampled from the shared observation table.

Next, a Kalman filter is used to maintain an estimate of the actual constraint between each pair of sensors over time. To achieve this, each new constraint (d_{ij},θ_{ij}) generated from the shared observations is converted to a Cartesian offset z_{ij} in sensor-relative coordinates, as in Eq. 1.

$$z_{ij} = \begin{bmatrix} d_{ij}\cos\theta_{ij} \\ d_{ij}\sin\theta_{ij} \end{bmatrix} \tag{1}$$

Using the latest measurements, the mean Cartesian offset $\hat{x}_{ij}$ and its variance $\sigma^2_{\hat{x}}$ are computed iteratively using Eq. 2-4. The measurement variance σ^2_z is assumed to be constant, and the mean and variance from the previous step are represented by x^-_{ij} and $\sigma^2_{x^-}$. K represents the Kalman gain.

$$K = \frac{\sigma^2_{x^-}}{\sigma^2_z + \sigma^2_{x^-}} \tag{2}$$

$$\hat{x}_{ij} = x^-_{ij} + K\left(z_{ij} - x^-_{ij}\right) \tag{3}$$

$$\sigma^2_{\hat{x}} = (1-K)\sigma^2_{x^-} \tag{4}$$

This procedure is repeated for each sensor pair, and the resultant offset estimates are converted to distance constraint d_{ij} and relative angle constraint θ^{rel}_{ij} for each pair of sensors.

F. Resolve constraints

To resolve this set of constraints over the entire set of sensors, we begin with a rough estimate of all sensor positions, and then refine those estimates using an iterative relaxation of constraints.

1) Initial position estimation

These constraint estimates are sorted by confidence, that is, in order of increasing variance. The highest-confidence sensor pair is chosen as the base sensor pair from which to estimate the other positions. We then iterate through the list and place subsequent sensors in relation to these base sensors in order of confidence.

2) Relaxation algorithm

The sensor position estimates are then refined, using an iterative relaxation algorithm. The algorithm and a proof of convergence are described in detail in [26]. In this algorithm, position estimates $(\hat{x}_{ji}, \hat{y}_{ji})$ are generated for each sensor i, relative to each neighboring sensor j located at position (x_j, y_j), according to Eq. 5. For these calculations, we use the estimated sensor angles to compute the angle offsets of the constraints in absolute coordinates: $\theta_{ji} = \theta_j + \theta^{rel}_{ji}$.

$$\hat{x}_{ji} = x_j + d_{ji}\cos(\theta_{ji}) \qquad \hat{y}_{ji} = y_j + d_{ji}\sin(\theta_{ji}) \tag{5}$$

Next, the variance v_{ji} of each position estimate is computed as the sum of the variance v_j of the neighbor sensor's position and the variance u_{ji} of the constraint estimate between the two sensors, as shown in Eq. 6.

$$v_{ji} = v_j + u_{ji} \tag{6}$$

Combining these position estimates, a new sensor position estimate (x_i, y_i) and its variance v_i are computed, according to Eq. 7. Here N_i represents the set of all sensors other than i.

$$\frac{1}{v_i} = \sum_{j\in N_i}\frac{1}{v_{ji}} \qquad x_i = \sum_{j\in N_i}\frac{\hat{x}_{ji}v_i}{v_{ji}} \qquad y_i = \sum_{j\in N_i}\frac{\hat{y}_{ji}v_i}{v_{ji}} \tag{7}$$

The angles of all sensors are then adjusted to minimize the average error between constraint angles and estimated sensor positions, using Eq. 8.

$$\theta_i = \arg\min\left(\sum_{j\in N_i}\left|(\theta_i + \theta^{rel}_{ij}) - \tan^{-1}\left(\frac{y_j - y_i}{x_j - x_i}\right)\right|\right) \tag{8}$$

This procedure can then be repeated until an arbitrary stopping point. In our experience the system tends to be sufficiently converged after three iterations.

G. Performance Evaluation

We tested the real-time performance of our calibration mechanism with a human tracking system in our lab. The sensor network used in these tests consisted of three SICK LMS-200 laser range finders covering an area roughly 6m by 4m, as shown in Fig. 7. Our tracking system is used in a variety of large and small environments, and we believe this is a reasonable size for the tracking coverage area in a room-sized indoor space.

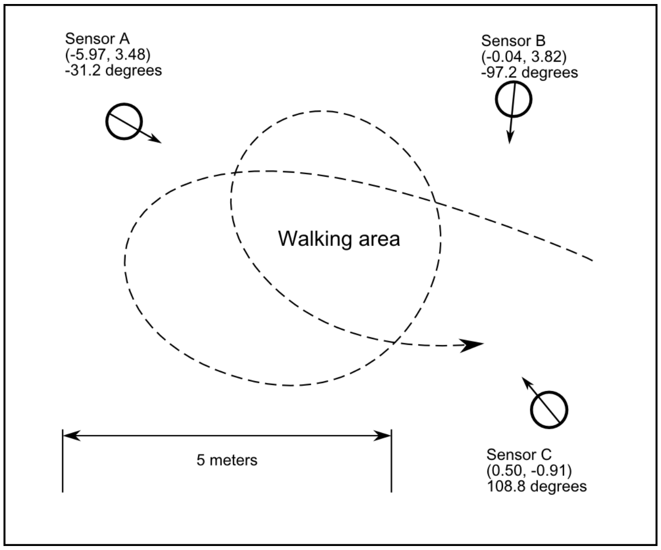

Fig. 7. Layout of sensors in performance evaluation space.

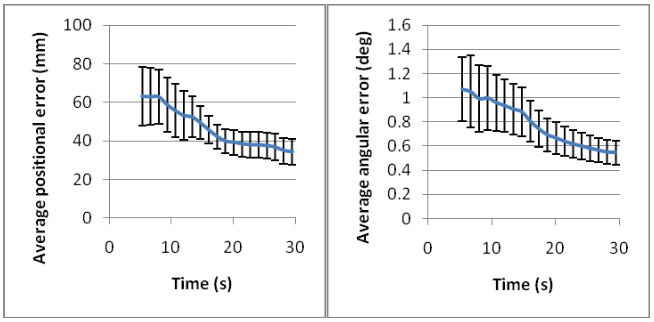

Fig. 8. Convergence of positional error (left) and angular error (right) over time, averaged over five trials. Error bars show standard error.

Before the evaluation, we performed calibration by hand to determine a ground truth for the sensor positions. One person was then instructed to casually walk a random path around the room while we calibrated the sensors. Five calibration trials were performed, with each trial lasting 30 seconds. Sensor position estimates were cleared between trials.

During calibration, data was collected for an initial burn-in period of 200 time steps (5.4 s), and then the sensor positions were estimated every 50 time steps (1.3 s) after that. Positional error and angular error for each pair of sensors were computed by comparing the distances and angles between the estimated sensor positions with the ground-truth values. These errors were then averaged across all sensor pairs to estimate overall calibration accuracy. Fig. 8 shows positional and angular error results averaged over all five trials.

These results show that within 10 seconds, our technique was able to estimate the sensor positions within 6 cm and orientations within 1° of accuracy. We consider this to be sufficient accuracy for our tracking systems in the field.

V. Error Detection

A. Error Detection Technique

When a sensor is physically moved during tracking, its detections of human positions are offset from those of other sensors, and its current field of view is also offset from its background scan. However, using only the sensor's immediate scan and background scan data, it is difficult to detect such misalignments.

We propose to use the human positions determined by the tracking algorithm at the previous time step to approximate the locations of the humans in the current scan. By masking out the scan lines which intersect these positions, we can isolate the portion of the current scan which should coincide with the background scan, as illustrated in Fig. 9-11.

Specifically, we have a set of N data points for each sensor, with values d_i ($1 \le i \le N$), represented in Fig. 9-10 by the solid lines, and a set of N background points b_i ($1 \le i \le N$), represented in Fig. 9-10 by the dashed lines. We also define a mask M to represent the set of all scan indices which intersect a human currently being tracked. The range of indices included in M is represented by the shaded boxes in Fig. 9-11.

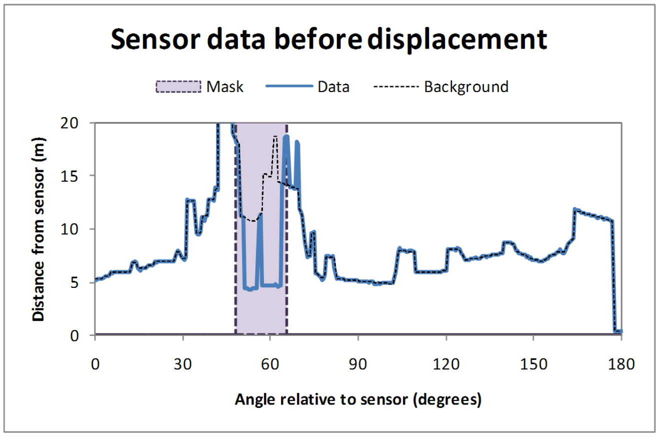

Fig. 9. Data and background scan for a properly-aligned sensor. Two people being tracked by the system are visible to this sensor, and a mask is generated to cover their estimated angular widths.

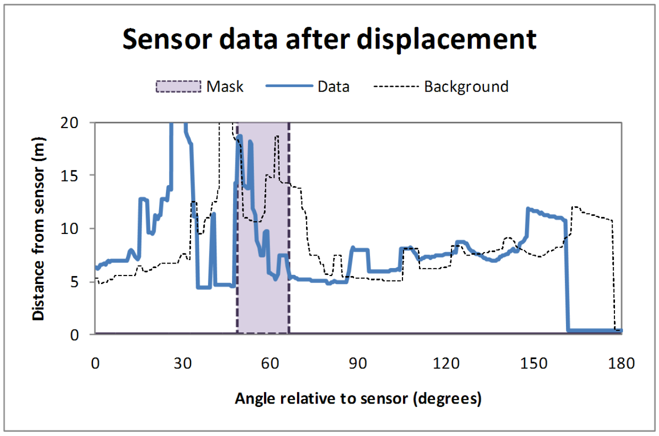

Fig. 10. Data and background scan for that sensor after a misalignment event. The mask and background scan are unaffected, but the real data is offset by a lateral offset of 10 cm and an angle of 15°.

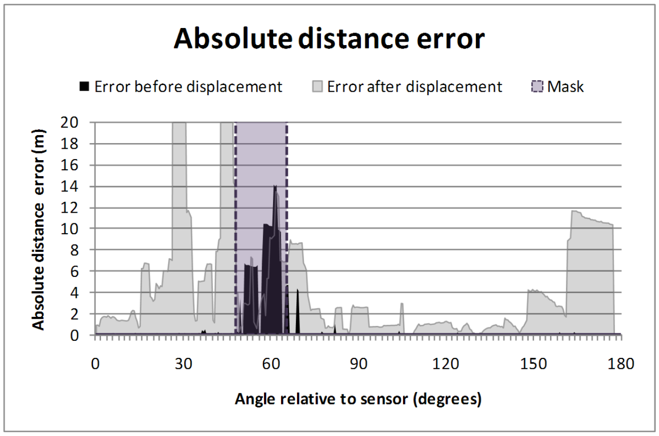

Fig. 11. The absolute distance error between sensor data and background scan, shown for the sensor positions shown in Fig. 7 and Fig. 8. Excluding the masked area, the average error value was 84 mm before displacement, and 5301 mm after displacement.

We use the absolute distance error between foreground and background scans, $|d_i - b_i|$, as an indicator of whether a sensor could be misaligned. The absolute distance error for the sensor *before* displacement, as a function of scan angle, is shown in black in Fig. 11. This indicates the difference between the two lines in Fig. 9. Here, the majority of the distance error can be attributed to the people being tracked, and thus the large values lie mostly within the masked region *M*. The function is very small outside of the masked region.

The absolute distance error *after* displacement is shown in gray in Fig. 11, representing the difference between the two lines in Fig. 10. This function has very large values outside the masked region, meaning those error values cannot be attributed to people being tracked. These values are large due to the offset between the background and foreground scans.

To isolate the error due to background scan offset from the error due to people, we define an error function *E* by summing the absolute distance error between the foreground and background scans over all points within the unmasked area, and normalizing this value by *N*, as specified in Eq. 9.

$$E = \frac{1}{N}\sum_{i=1}^{N}\begin{cases} |d_i - b_i| & i \notin M \\ 0 & i \in M \end{cases} \tag{9}$$

Our error detection algorithm sets a sensor's error flag when the value of *E* exceeds an error threshold. When an error is flagged, the system automatically removes the sensor's data from the tracking calculations. It also alerts a human operator to rebuild that sensor's background scan and re-calibrate its position. This process could be automated, but we prefer to alert the operator because the sensor may need to be physically moved back into position.

B. *Performance Evaluation*

We evaluated our error detection system in two ways: first, through simulations using recorded data from our field trials, and second, through real-time trials using a human tracking system inside our laboratory.

1) *Simulation*

To evaluate our error detection mechanism in a high-traffic environment, we performed tests using data from our Universal CityWalk field trial environment on busy days. Since actual instances of sensors being pushed out of alignment were not logged, we simulated the displacements.

To simulate a displacement, a sensor's estimated position was offset by a random distance (Gaussian, σ=100 mm) and angle (flat distribution, $-45^\circ < \theta < 45^\circ$), and its scan data were translated, rotated, and resampled to simulate the offset.

1000 displacement events were simulated in a 3-hour data set, and *E* was recorded for each sensor before and after each event. The distributions of error values for properly-aligned and displaced sensors are shown in Fig. 12.

This comparison enables an appropriate threshold value to be determined on the basis of minimizing false alarms (type I errors) or missed alarms (type II errors). For example, choosing a threshold value of 350 would lead to a 0.8% false alarm rate and a 2.6% missed alarm rate for this data set.

Using this threshold, we verified the performance of our error detection system using a different 3-hour data set. Of 1000 post-displacement and 1000 pre-displacement evaluations, there were 17 missed alarms (1.7%) and only 2 false alarms (0.2%).

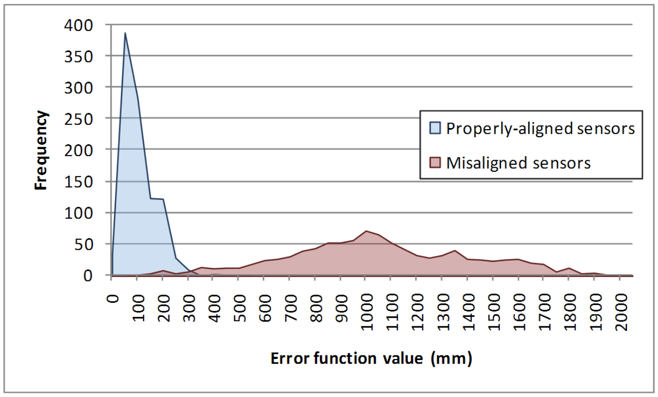

Fig. 12. Frequency histogram of error function values for properly-aligned and displaced sensors in our simulation using recorded data.

2) *Online Performance*

To validate this technique with an online system, we used the 3-sensor human tracking system described in Sec. IV-G. One person was asked to casually walk a random path around the room, while another was asked periodically to physically push one of the sensors out of place to an approximate lateral displacement of up to10 cm, and an angular displacement of up to 45°.

Before and after each displacement we recorded the error value *E* for the displaced sensor. After each displacement, the sensor's background scan was rebuilt and its position re-calibrated.

Over 10 trials, the average value of *E* was 240 before sensor displacement and 1160 after displacement. The system correctly detected all 10 displacements, and there were no false alarms.

VI. Discussion

A. *Stopping criteria*

Although a formal stopping criterion has not been defined in this paper, the estimates of sensor variance give good indications of convergence of the system. However, we have not formally tested the robustness of this stopping criterion.

B. *Absolute position calibration*

The technique in this paper addresses the problem of relative sensor position calibration, but registration with an external coordinate system will usually be necessary once relative calibration has been completed.

Our system includes a graphical interface that allows the calibrated sensor network to be manually dragged and rotated relative to a fixed map of the tracking area in absolute coordinates. To obtain a first approximation before this manual registration step, a list of rough sensor positions can be used to define the starting position for the base sensor pair during position estimation.

C. *Field-of-view overlap*

Generally speaking, it is not necessary to establish constraints between every pair of sensors. The minimum requirement is that there be some chain of constraints eventually linking any pair of sensors, *i.e.* if constraints A-B and B-C are known, computation of constraint A-C is not necessary.

Because these constraints are approximate, however, calibration accuracy increases as more constraints are known. Since constraints can only be computed from shared observations, best performance is obtained when there is a large field-of-view overlap between sensors.

D. Time Delay

In our laboratory system as well as our field system, data from all sensors was collected on a single PC through a wired connection, so network lag and clock synchronization issues are not a concern.

However, for systems using wireless networks or multiple data collection PC's, synchronization is an important issue. By using NTP to synchronize PC clocks and tagging sensor data with time stamps, it should be possible to synchronize the data from multiple sensors and perform sensor calibration even when there are network delays.

VII. Conclusions and Future Work

We have proposed a technique for automatically determining relative positions and orientations for laser range finders by observing only pedestrian motion in the space, and without using scan matching or explicit landmarks. We have also proposed a method for determining when any sensor in the network has been displaced, information which can be used to prevent corrupted data from being used in tracking calculations and to notify an operator of the problem.

We have demonstrated our calibration technique in real time with a tracking system in the laboratory, and we have shown the effectiveness of our displacement detection technique both in laboratory tests and in simulations based on recorded data.

References

[1] A. Fod, A. Howard, and M. Matarić, “A laser-based people tracker,” *Proc. IEEE International Conference on Robotics and Automation*, pp. 3024–3029, Washington DC, May 2002.

[2] A. Panangadan, M. Matarić, and G. Sukhatme, “Detecting anomalous human interactions using laser range-finders,” *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2136–2141. IEEE Press, Sep 2004.

[3] H. Zhao and R. Shibasaki, “A novel system for tracking pedestrians using multiple single-row laser-range scanners,” *IEEE Transactions on Systems, Man, and Cybernetics* 35, 283 (2005).

[4] J. Cui, H. Zha, H. Zhao, R. Shibasaki, “Laser-based detection and tracking of multiple people in crowds,” *Computer Vision and Image Understanding (CVIU)*, Vol. 106, Issues 2-3, pp. 300-312. May 2007.

[5] E. Prassler and J. Scholz and P. Fiorini, “Navigating a robotic wheelchair in a railway station during rush hour,” *Int. Journal on Robotics Research*. 18: 760-772, 1999.

[6] M. Montemerlo, S. Thun, W. Whittaker, “Conditional particle filters for simultaneous mobile robot localization and people-tracking,” in *Proc. IEEE International Conference on Robotics & Automation (ICRA)*, 2002.

[7] D. Schulz, W. Burgard, D. Fox, and A. B. Cremers, “People tracking with mobile robots using sample-based joint probabilistic data association filters,” *International Journal of Robotics Research*, 22(2):99–116, 2003.

[8] T. Kanda, D. F. Glas, M. Shiomi, H. Ishiguro and N. Hagita, Who will be the customer?: A social robot that anticipates people's behavior from their trajectories, *Tenth International Conference on Ubiquitous Computing (UbiComp 2008)*, pp.380-389, 2008.

[9] D. F. Glas, T. Kanda, H. Ishiguro, N. Hagita, “Simultaneous people tracking and localization for social robots using external laser range finders, *Proc. 2009 IEEE International Conference on Intelligent Robots and Systems (IROS 2009)*

[10] T. Ikeda, H. Ishiguro, D. F. Glas, T. Miyashita, N. Hagita, “Person Identification by Integrating Wearable Sensors and Tracking Results from Environmental Sensors,” *IEEE International Conference on Robotics and Automation (ICRA 2010)*

[11] S. Satake, T. Kanda, D. F. Glas, M. Imai, H. Ishiguro, N. Hagita, “How to approach humans? - strategies for social robots to initiate interaction,” *4th ACM/IEEE International Conference on Human-Robot Interaction (HRI2009)*, pp.109-116, 2009.

[12] D. F. Glas, T. Miyashita, H. Ishiguro, and N. Hagita, “Laser-based tracking of human position and orientation using parametric shape modeling,” *Advanced Robotics*, Vol. 23, No. 4, pp. 405-428, 2009.

[13] J. Hightower and G. Borriello, “Location systems for ubiquitous computing,” *IEEE Computer*, vol. 34, no. 8, pp. 57-66, August 2001.

[14] P. J. Besl and N. D. McKay, “A method for registration of 3-D shapes,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239-256, 1992.

[15] A. Diosi and L. Kleeman, “Laser Scan Matching in Polar Coordinates with Application to SLAM,” in *Proc. of 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'05)*, August, 2005, Edmonton, Canada.

[16] G. Weiss, E. Puttkamer, "A map based on laser scans without geometric interpretation," *Intelligent Autonomous Systems* vol. 4, pp. 403- 407 (1995).

[17] T. Röfer, “Using histogram correlation to create consistent laser scan maps,” in *Proc. IEEE International Conference on Robotics Systems (IROS '02)*, pp. 625-630, Lausanne, Switzerland.

[18] N. Ripperda and C. Brenner, “Marker-free registration of terrestrial laser scans using the normal distribution transform,” *Proceedings of the ISPRS Working Group V/4 Workshop 3D-ARCH 2005*, Mestre-Venice, Italy , 2005

[19] P. Biber, W. Straßer, “nScan-matching: simultaneous matching of multiple scans and application to SLAM,” *Proc. 2006 IEEE International Conference on Robotics and Automation (ICRA '06)*, pp. 2270-2276.

[20] D. Fox, W. Burgard, F. Dellaert, and S. Thrun, “Monte Carlo localization: efficient position estimation for mobile robots,” *Proc. 16th National Conference on Artificial Intelligence (AAAI'99), 1999.*

[21] T. Sogo, “Localization of sensors and objects in distributed omnidirectional vision,” Ph.D. dissertation, Department of Social Informatics, Kyoto University, Kyoto, Japan, 2001.

[22] Y. Shang and W. Ruml, “Improved MDS-based localization,” *Proc. IEEE INFOCOM '04*, Hong Kong, 2004.

[23] N. Krahnstoever and P. R. S. Mendonca, “Bayesian autocalibration for surveillance,” *Proc. Tenth IEEE International Conference on Computer Vision*, 2005.

[24] I. Junejo, H. Foroosh, “Robust auto-calibration from pedestrians,” *Proc. IEEE International Video and Signal Based Surveillance (AVSS '06),* 2006.

[25] K. O. Arras, O. M. Mozos, W. Burgard, "Using boosted features for the detection of people in 2D range data," *Proc. 2007 IEEE International Conference on Robotics and Automation (ICRA'07)*, pp 3402–3407, Rome, Italy, 2007

[26] T. Duckett, S. Marsland, and J. Shapiro, “Learning globally consistent maps by relaxation,” in *Proc. IEEE International Conference on Robotics & Automation (ICRA)*, San Francisco, CA, April 2000.