The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

Simultaneous People Tracking and Localization for Social Robots Using External Laser Range Finders

Dylan F. Glas, Takayuki Kanda, *Member, IEEE*, Hiroshi Ishiguro, *Member, IEEE*, Norihiro Hagita, *Senior Member, IEEE*

***Abstract*—Robust localization of robots and reliable tracking of people are both critical requirements for the deployment of service robots in real-world environments. In crowded public spaces, occlusions can impede localization using on-board sensors. At the same time, teams of service robots working together need to share the locations of people and other robots on the same global coordinate system in order to provide services efficiently.**

To solve this problem, our approach is to use an infrastructure of sensors embedded in the environment to provide an inertial reference frame and wide-area coverage. Based on a people-tracking system we have previously established which uses laser range finders to track people's trajectories, we have developed a technique to localize a team of service robots on a shared global coordinate system. Each robot's odometry data is associated with the observed trajectory of an entity detected by the laser tracking system, and Kalman filters are used to correct rotational offsets between the robots' individual coordinate systems and the global reference frame. We present our data association and pose correction algorithms and show results demonstrating the performance of our system in a shopping arcade.

I. Introduction

RESEARCHERS around the world are developing robots to interact with people within real social environments, such as museums [1], hospitals [2], and schools [3]. Although basic navigational tasks like obstacle avoidance are important for these robots, the navigation-related research in this field tends to focus on psychological aspects of human-robot interaction such as motion strategies appropriate to social environments [4] and human-aware motion planning [5].

At the same time, other researchers are developing highly-accurate and robust navigation and mapping techniques for robots which will operate in explorational and military contexts. Some of these techniques are based on fixed references such as GPS [6] or external landmarks [7], and others are based on visual or RFID tagging of the robot or environment [8], or integration with intelligent environments [9]. Perhaps one of the most common navigational techniques is to use on-board laser scanners, possibly combined with vision, for localization and mapping.

While mapping-oriented localization techniques are valuable and essential for a wide range of exploratory and military applications, the requirements for such systems differ somewhat from the requirements of service robots designed for social interactions with people in crowded public spaces.

Our research aim is to develop a technology framework enabling social robots to interact with people, provide services, and collaborate with other robots in real social spaces such as shopping malls or convention centers, like the example shown in Fig. 1. These services might include things like offering directions to customers who seem to be lost, or offering shop recommendations to customers who appear to be interested.

Figure 1. Robots provide route guidance and shop recommendations to customers in one of our field trials at a shopping arcade.

For this type of application, it is essential for the robots to be reliably localized in a shared coordinate system, to allow them to collaborate with each other and avoid collisions. In addition, large numbers of people passing through the space may crowd around the robots, which means that the robots must be able to navigate in crowded environments with many occlusions. It is also necessary for the robots to be able to track people in the environment robustly and continuously, for example, to avoid two robots successively approaching the same person to offer the same service.

II. Related Work

In the robot navigation community, much attention has

Manuscript received March 1, 2009. This research was supported by the Ministry of Internal Affairs and Communications of Japan.

D. F. Glas, T. Kanda, 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@ams.eng.osaka-u.ac.jp).

been devoted to the problem of simultaneous localization and mapping (SLAM) using onboard sensors [10], [11]. This technique serves the dual purpose of map-building and robot localization. However, for robots performing services in a defined commercial space, the map-building functionality serves no useful purpose aside from its role in aiding localization.

Additionally, in busy public spaces, the presence of large numbers of people presents a challenge for tasks such as map-matching, as people crowded around the robots can occlude the fixed references that would be used for map-matching. Fig. 2 shows example data from laser range finders located around our environment, indicating the degree of crowdedness that such a system will need to accommodate.

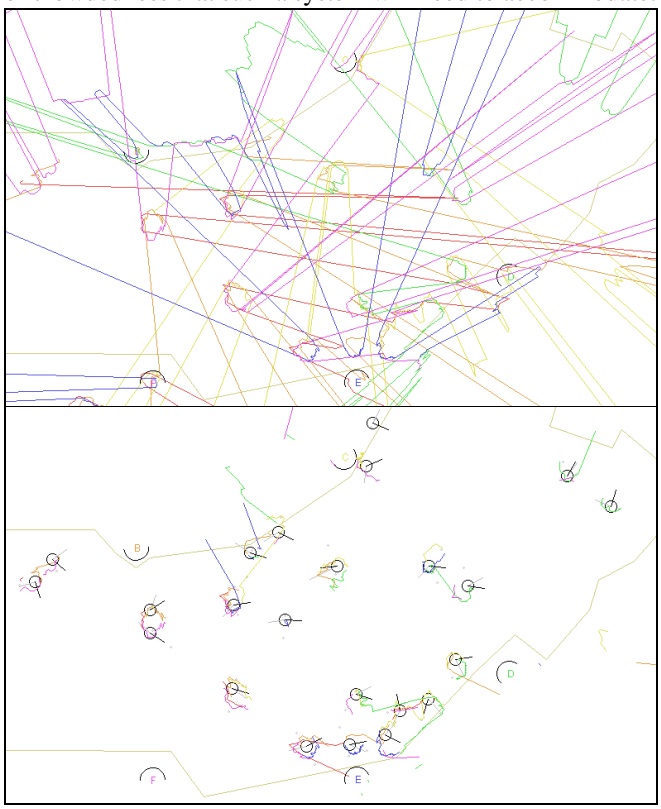

Figure 2. Scan data from six laser range finders in our field trial space on a busy day. (Top) Raw laser scan data. (Bottom) Scan after processing and background removal, including locations of detected people.

Regarding robot navigation in crowded spaces, Prassler *et al.* demonstrated a robotic wheelchair able to operate in extremely crowded environments, but their concern was with obstacle avoidance, not localization [12]. Robots such as Rhino, Minerva, and GRACE have demonstrated Markov localization in crowded environments, but these systems did not track people individually [13], [14].

The task of localization in conjunction with people tracking is an active topic of research and some techniques addressing the problem have been published; however, these works generally deal with small numbers of people. For example, Montemerlo *et al.* demonstrated such a system with up to 5 people simultaneously [15]. A technique for differentiation between moving people and static elements of the environment was also developed by Wolf *et al.* [16], but demonstrated with only three moving people in the environment. Schulz *et al.* demonstrated a people-tracking technique with four people simultaneously [17].

Related research using road vehicles has focused on combining SLAM with the detection of moving objects in urban environments, such as [18] and [19].

Collaborative SLAM with multiple robots is also an active field of research [20], [21], however research in this field tends to focus on exploring optimal search techniques, for example, to minimize mapping time. In our case, exploration is not of interest, so such optimizations are unnecessary.

Our approach uses fixed external laser range finders rather than the on-board sensors of the robots. A similar approach using external cameras is presented in Ref. [22].

III. Network Robot Platform

A. Architecture Overview

We developed a technology platform to enable us to study social human-robot interaction with multiple robots in a real-world setting. In our design, an environmental monitoring system using laser range finders tracks the motion of people through a space where several service robots are operating.

The robots are connected via wireless networks to a tracking server, a planning server, and a teleoperation system, as shown in Fig. 3.

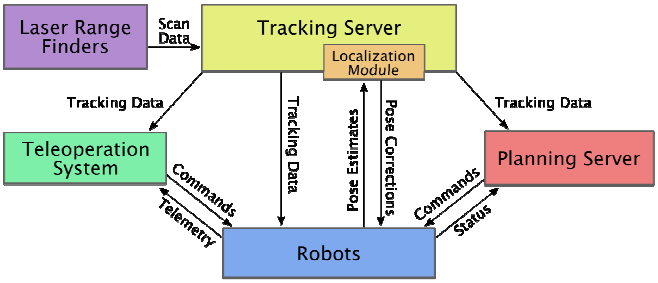

Figure 3. System architecture for the network robot platform. The tracking server uses laser range finder data to track people and robots in the environment. This data is then used for path planning, service allocation, robot localization, and operator supervision.

B. Robots

For most of our experiments and demonstrations, we used four humanoid Robovie II communication robots [23]. The robots are semi-autonomous and perform local path planning tasks on their own, such as avoiding obstacles or approaching people. The robots receive commands from the planning server or from an operator via the teleoperation system, specifying which individuals to approach and what services to perform.

Once the robot has successfully approached a person, the robot handles the social interaction according to its internal rules and behaviors, with no assistance from the planning server. During interactions, a remote operator assists the robot with speech recognition via the teleoperation system.

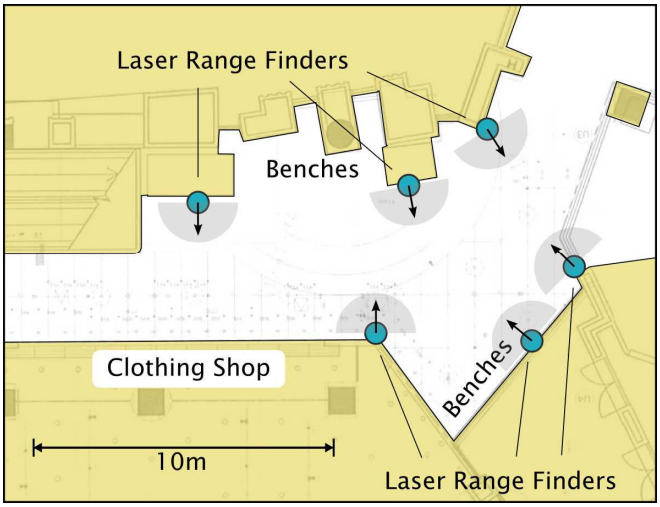

Figure 4. Operational environment for our field trials. Six laser range finders monitored the central area, where robots approached customers and offered directions and recommendations.

C. Tracking Server

The tracking server processes the laser range finder data to track humans and robots in the environment. This information is used by the planning server to assign robot services and perform path planning, and it is used by the teleoperation system to enable the operator to supervise the robots. Information about human positions is also used by the robots themselves for planning local trajectories for approaching or avoiding people. Finally, a localization module in the tracking server receives odometry data from each robot and sends back pose corrections based on tracking data.

D. Planning Server

The planning server uses characteristics of people's trajectories and a statistical analysis of movement patterns to identify individuals to whom the robots should offer services, and it assigns individual robots to approach them [24]. The planning server also provides high-level path-planning for the robots, so the robots are coordinated with each other and do not need any overall knowledge or maps of the area.

E. Teleoperation System

Our system includes a teleoperation console, from which an operator can supervise several robots simultaneously. In our applications, the operator's primary roles are to support speech recognition in noisy environments and to generally supervise the robots for safety. For many of our demonstrations and experiments, the operator has also corrected failures in robot localization [25]; however, the localization system presented here features global error recovery in the case of tracking failures, enabling the operator to focus completely on conversation-related tasks.

F. Operating Environment

We installed our system in an open space roughly 20 m long by 5 m wide in Universal CityWalk Osaka, a shopping arcade in front of the Universal Studios Japan theme park. Within this space we simultaneously operated four humanoid robots for a number of demonstrations and experiments. During the demonstrations, it was common for more than 30 people to be present in the space, often crowding around the robots to interact with them.

Six laser range finders were installed in the space for human and robot tracking (see Fig. 4). These were arranged around the perimeter of the space to minimize occlusions. The data from these sensors was recorded on a data-acquisition PC and streamed over the network to the tracking server.

IV. Localization System

A. Tracking with Laser Range Finders

In this environment, we used a network of six SICK LMS-200 laser range finders, positioned around the perimeter of the area. They were set to a detection range of 80 m with 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, a height chosen so the sensors could see above clutter and obstacles such as benches and luggage. Another reason for this placement was that at long range, the scan beams are spaced quite far apart (over 8 cm apart at a range of 10 m) and detection of small features like legs is difficult. Detection of larger targets, like a torso, is more robust at these distances.

The algorithm we used for detecting and tracking humans is described in [26], but we will summarize the relevant parts of the algorithm here.

1) Detecting Entities

To identify new entities (humans or robots), the raw scan data is segmented at every time step, to extract continuous segments of foreground data roughly corresponding to expected entity widths (in our case, robots and humans are roughly the same width). Clusters of these patterns are grouped together and flagged as candidate entities. Candidates coinciding with entities already being tracked are removed from the list, and those remaining are propagated to the next time step, where they are merged with the candidates detected during that step.

If a candidate entity survives beyond a threshold number of time steps, it is considered to be a valid detection, and a new particle filter is assigned to that location, initialized with the position and velocity of the candidate entity it replaces. At this point, no distinction is yet made between humans and robots, and all entities are initially assumed to be humans.

2) Tracking Entities

In our algorithm, each new entity is assigned a unique ID and tracked by an individual particle filter. By doing so, we allow feature-object associations to be handled implicitly by the particle filters, which follow the detected features over time, rather than requiring re-association between observations and previously observed entities at every time step. This approach generally produces reliable results,

although entities are sometimes lost briefly and redetected with a new ID.

By contrast, the associations between robots and detected entities *are* explicitly handled at each time step, because reliable localization is critical to the system's performance and justifies the additional computational overhead.

3) Removing Entities

The removal process is much simpler than the addition process. When the particles within a filter spread out beyond a defined dispersion threshold, or when their average likelihood value goes below a threshold probability, that particle filter is assumed to no longer be tracking anything, and it is removed.

B. Particle Filter Design

We assume that the reader is familiar with particle filters, a common sample-based technique often used for Bayesian state estimation in robotics. For a basic explanation of particle filtering and related techniques, see Ref. [27].

In our implementation, particle filters were used to estimate four state variables (x,y,v,θ) for each entity being tracked. In the resampling step of the filter, we used the Sampling Importance Resampling method. To minimize the number of particles, we used the KLD-sampling technique [28].

The key elements which define the behavior of a particle filter are the motion model used for propagating the particles and the likelihood model used for assigning weights to them.

1) Motion Model

As has been observed in [29], the modeling of human motion presents difficulty because it is neither Brownian in nature, nor can it be modeled as a smooth linear function, since people may stop or change direction abruptly. Thus, as a compromise between the two, a Gaussian noise component is added to each particle's v and θ values to capture the randomness of human motion.

2) Likelihood Model

Laser scan data provides two qualitatively distinct types of information useful for estimating human positions: occupancy information, indicating whether a certain point is occupied or empty, and edge information, indicating a contour which may correspond with the edge of a detected object. Fig. 5 illustrates the distinction between these two kinds of information.

To determine likelihood values from the raw sensor data, our system uses an adaptive background model, updated over time to determine the best estimate of the true background distance. Occupancy likelihood is determined by dividing the world into "open", "shadow", and "unobservable" regions. The "unobservable" region is beyond the background model for that sensor, and thus contributes no information. The "open" region has been observed by the sensor to be unoccupied, and the remaining space is considered "shadow". Note also that every "shadow" region lies behind an "edge".

For Bayesian state estimation, it is necessary to model the conditional probability $p\left(z_t \mid x_t^{[m]}\right)$ for observation z_t and state hypothesis $x_t^{[m]}$ for particle m at time t.

Figure 5: A typical single-sensor laser scan. (Left) The positions of humans relative to the scanner. (Center) Occupancy information. (Right) Edge information.

Our likelihood model for this calculation is expressed in Eq. 1 and 2 and includes components reflecting both occupancy and edge information.

$$p\left(z_t \mid x_t^{[m]}\right) = \sum_{i=1}^{n_{sensors}} p_i\left(z_t \mid x_t^{[m]}\right) - p_{collocation} \quad (1)$$

$$p_i\left(z_t \mid x_t^{[m]}\right) = \begin{cases} p_{shadow} + p_{edge}\left(z_t \mid x_t^{[m]}\right) & \mid shadow \\ p_{open} & \mid open \end{cases} \quad (2)$$

For a point in a shadow region (strictly speaking, we consider only those regions wide enough to contain a human), the likelihood in Eq. 2 is calculated as the sum of a constant value p_{shadow} and a likelihood $p_{edge}\left(z_t \mid x_t^{[m]}\right)$, calculated as a normal distribution centered upon a point located one approximate human radius behind the observed edge. In our calculations a value of 25cm was used for humans, and a robot-specific radius was used for each robot.

This model reflects the fact that people are highly likely to be found just behind an edge, yet can plausibly exist anywhere in a shadow region (e.g. the occluded person in Fig. 5). For a point in an open region or a shadow region too narrow to contain a human, the likelihood is theoretically zero, but for robustness to noise we set it to a small but nonzero constant value p_{open}. In this case, edge information is irrelevant.

Finally, in Eq. 1, these likelihood values are averaged across all $n_{sensors}$ sensors for which the proposed point lies within the sensor's "open" or "shadow" range, i.e. not "unobservable" to that sensor. To prevent two particle filters from tracking the same entity, a value $p_{collocation}$ is subtracted from this result, calculated as a sum of normal distributions surrounding each of the other entities, based on the list of human positions from the previous time step.

C. Association Algorithm

As mentioned above, a separate algorithm is used to maintain the correspondence between robot positions and detected entities.

The tracking server internally maintains a tracking model for the position, velocity, and angular velocity of each connected robot. When each robot reports its estimated wheel velocities, the tracking server updates the linear and angular velocities for that robot's tracking model.

In parallel with these updates, after each iteration of the particle filters in the tracking system, the associations between all robots and detected entities are updated according to the

following algorithm:

1) Project robot pose

For each robot tracking model r_t, calculate a kinematic projection of the robot's pose based on angular velocity, linear velocity, and the elapsed time since last update.

2) Remove invalid associations

Our tracking system is quite stable, so it is uncommon for associations to be changed while the robots are in operation. However, two kinds of errors can happen. Local tracking errors occur when there is an error in the laser-tracking system, such as when the particle filter tracking the robot disappears briefly. Global tracking errors occur when the robot is mistakenly associated with the wrong entity entirely, in the wrong part of the environment.

To handle these errors, we first remove the association between r_t and its associated entity if any of the following conditions are true:

(a) the associated entity no longer exists

(b) r_t is too far from the entity

(c) velocity of r_t is too different from velocity of entity

(d) another robot is associated to this entity

3) Identify possible new associations

Next, in order to generate a new association for the robot, we need to create a list of entities which are potential matches. By comparing the robot's current velocity and motion history with those of each entity being tracked, we are able to identify which entities are *unlikely* to be tracking the robot.

For this step, the system keeps a history of absolute distance traveled for each robot (according to odometry) and each entity tracked in the environment. The history is updated once per second. By iterating over this history and calculating the root-mean-square (RMS) error between the motion history of a robot and each tracked entity, we can identify which entities are not likely to be tracking the robot's motion.

Fig. 6 shows an example of this motion history taken from our field trial, and Fig. 7 shows the RMS difference between the motion history of each entity and the odometry-based motion history of the robot for the same dataset.

Note that this information is not guaranteed to uniquely identify a best match. For example, when robots and people are stopped for extended periods of time, it is not possible to distinguish between them using motion history alone.

This technique is useful, however, in filtering out entities which are *not* possible matches for the robot. By removing those entities from the list of all entities being tracked, we generate a short list of potential matches for each robot. This list is updated only once per second, and thus does not incur a significant computational overhead.

In addition to this filtering, the difference between the robot's current speed and the current speed of each remaining entity is calculated, and in cases where the difference between the speeds exceeds a threshold (we used 350 mm/s), we remove that entity from the list of potential matches as well.

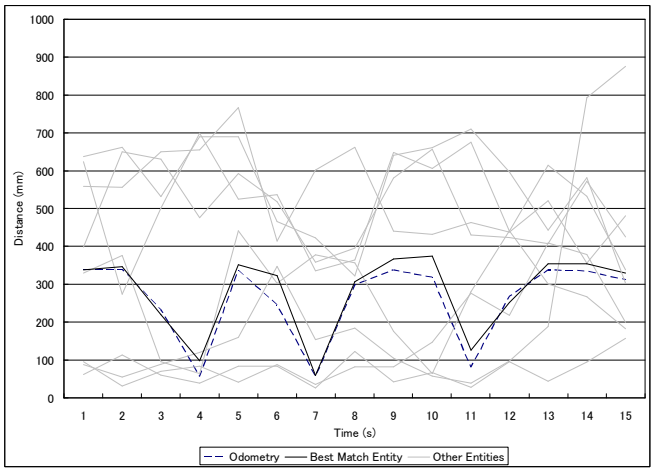

Figure 6: Motion history. Total distance traveled per second over a period of 15 seconds. Solid lines show the motion histories for nine entities detected by the laser tracking system. The dashed line shows the robot's reported odometry data, and the solid black line indicates the motion history for the entity corresponding to that robot.

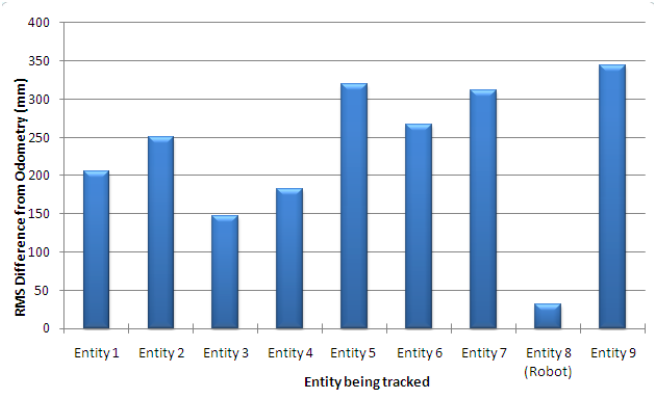

Figure 7: RMS difference between distance traveled by tracked entities and distance reported by robot's odometry, taken over a 15-second history window.

4) Associate robots to entities

There are two methods we use for re-associating robots to tracked entities, corresponding to the two kinds of tracking errors which can occur.

To correct local tracking errors, we use a *local association* algorithm. From the set of all plausible matches, we consider only those entities which are within a threshold distance from the robot's current estimated position (we used a distance of 1 m). We then associate the robot with the entity closest to its projected position, if any candidates remain.

If a robot is not associated with a plausible match within five seconds, we apply a *global association* algorithm. In this case, we evaluate an extended motion history vector, covering up to 300 s. The RMS error between each plausible entity and the robot is calculated over this vector, giving a more accurate estimation of the best match for that robot, and allowing more entities to be ruled out.

For further disambiguation, this list is prioritized by similarity of the entity's current speed to the current speed of the robot. Thus, for global association, distance from the robot's estimated position is not considered, allowing the

robot to be re-associated anywhere within the environment.

Finally, we must consider the case when two robots are mapped to the same entity. When this happens, the robot with the closest velocity to the tracked entity is given priority, and the other robot is re-associated to its next best global match.

5) Apply position corrections

For each robot which has been successfully associated to a tracked entity, we use the position data from the laser tracking system to correct the robot's estimated position.

If the robot is moving, we set the position of r_t to the latest observed position of its associated entity.

If the robot is stopped, we add the latest observed position to a window filter. This enables the positions to be averaged over time, resulting in a more stable position estimate for stationary robots.

6) Update output data

Finally, we incorporate the resultant robot positions into the output list of human and robot positions which is sent to the planning server. Position corrections are also sent to the localization module, where they are used in the angle correction calculations and then sent to the robots.

D. Theta Correction

Simple lateral position errors will converge over time, as the corrected position estimates sent to the robots approach their actual positions. However, angular errors can increase over time with odometry drift, and even small angular errors in the robot's orientation will lead to systemic position errors. If the angular errors are large enough, this can lead to an increased incidence of false disassociations and instability of the tracking system. Thus a mechanism is required to correct orientation errors.

Since the position estimate is already calculated for us by the tracking server, the only remaining variable requiring correction is the robot's orientation angle θ. We employed a one-dimensional Kalman filter for this calculation.

Kalman filtering is another common state-estimation technique, in which both an estimate of the state of a system and a measure of the accuracy of that estimate are calculated recursively at every time step. There are several forms of the equations for the Kalman filter, and Equations 3-5 show the form which most clearly illustrates our approach.

Our aim is to recursively approximate the actual angle θ_k at time step k with an estimate $\hat{\theta}_k$, given angle observation z_k with measurement variance $\sigma^2_{z_k}$ and previous angle estimate θ^-_k. The variance $\sigma^2_{\theta^-_k}$ of the angle estimate is updated at each step, and it is assumed that this increases over time due to a process noise with variance $\sigma^2_{process}$. The term K_k in Eq. 3 represents the Kalman gain, which is used in calculating the new angle estimate (Eq. 4) and its variance (Eq. 5). As each new observation is recorded, these three equations are evaluated recursively, using the previous step's state estimate and variance.

$$K_k = \frac{\sigma^2_{z_k}}{\sigma^2_{z_k} + \sigma^2_{\theta^-_k}} \tag{3}$$

$$\hat{\theta}_k = \theta^-_k + K_k\left(z_k - \theta^-_k\right) \tag{4}$$

$$\sigma^2_{\hat{\theta}_k} = \sigma^2_{\theta^-_k} + K_k\left(\sigma^2_{\theta^-_k}\right) + \sigma^2_{process} \tag{5}$$

The state being estimated in this case is the robot's orientation angle. The tracking system is unable to observe the robot's angle explicitly, so we are forced to rely on observations of the robot's direction of motion to measure the robot's angle, as illustrated in Fig. 8.

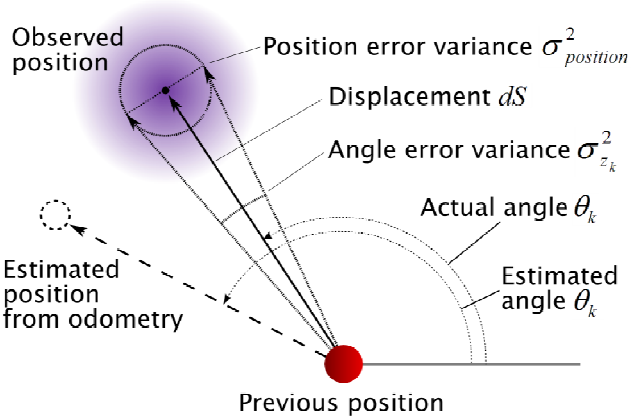

Figure 8. Dynamic model showing estimated pose, observed pose, and covariance of measurement error.

The Kalman filter alternates between two phases: a prediction phase and an update phase.

Prediction: Beginning with the robot's previous position (which has already been corrected to coincide with the observed position from the previous time step), we predict the robot's new position based on odometry. We assume that this motion is corrupted with a small process noise $\sigma^2_{process}$ at each step (we used an arbitrary small value of 0.0001 radian).

Update: The position data from the laser tracking system is used to provide an observed value for the robot's angle. The position observation is noisy, with a measurement uncertainty $\sigma^2_{position}$. Using a small-angle approximation, we can say that for displacement dS, the variance of the angle error $\sigma^2_{z_k} \approx \frac{\sigma^2_{position}}{dS}$.

What this means is that the angle estimate from the tracking system becomes more reliable (lower variance) when the robot has traveled further (higher dS), which makes intuitive sense.

Each time a new position estimate is available from the laser tracking system, these two steps are repeated and the difference between the angle estimate $\hat{\theta}_k$ and the robot's perceived angle is calculated and sent to the robot.

E. Network Latency

Because this system operates over a wireless network, data loss and delay due to interference and signal loss is a common problem. In particular, delayed data can cause old position corrections to be applied to a robot after it has been updated to a new position. This can result in the robot's perceived position jumping between the new and old position, as old and new data cycles between the robot and the localization server.

To eliminate this problem, whenever the robot's position is set to a new location (as opposed to an incremental position correction) a "reset" signal is sent from the robot through the localization system, and back to the robot. Until the signal returns, the robot does not respond to any position corrections.

V. Performance Evaluation

A. Theta Correction

To demonstrate the effectiveness of the theta correction system, we ran two robots simultaneously for 7 minutes in the field trial environment, recording tracking and odometry data. We then replayed the data through the localization system twice, once with theta correction on and once with it off, to observe the angular error which would have occurred without theta correction. Fig. 9 shows results from this comparison.

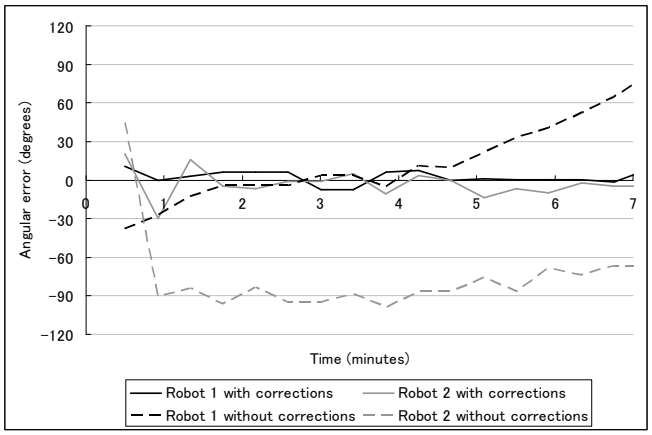

Figure 9: Angular error over time with and without theta correction. This graph shows the average difference between a robot's perceived orientation angle and its motion direction, evaluated every 30 seconds for 7 minutes.

In this trial, the robots patrolled paths of 2-3 m in length, stopping to interact with people only when the customers actively approached the robots. To avoid any periodic effects of the repetitive patrol path, each robot was also commanded once to drive to a point further than 5 m away before returning to its patrol path autonomously. The robots each traveled approximately 55 m and turned through an absolute angular displacement of at least 8 rotations. During this trial an average of 6.7 people were tracked in the environment at any given time, with a maximum of 14 tracked simultaneously.

Two kinds of errors are visible in the uncorrected robot angle estimates in Fig. 9. The angle estimate for Robot 1 was quite close to its actual orientation for several minutes, but eventually drifted away, most likely due to wheel slip during the turns in its patrol path. Robot 2 does not appear to have had significant drift due to wheel slip, but its initial angle is around 90 degrees off from its actual direction of motion, and this error remains uncorrected for the duration of the trial.

B. Tracking Performance

Stability in tracking is also a key requirement of this system. To evaluate tracking performance, we ran four robots simultaneously in the field trial environment on a moderately crowded day for 30 minutes (roughly the amount of time we can continuously run the robots before changing the batteries) and recorded the tracking data for that period. We analyzed the data and manually identified localization errors.

Fig. 10 shows the results of this analysis and the number of people being tracked during that period. There were four instances of localization errors during this time, during each of which only one robot was not correctly localized. Each error was automatically corrected after a short time. Note that since four robots were operating simultaneously, the system correctly tracked the robots more than 98% of the time.

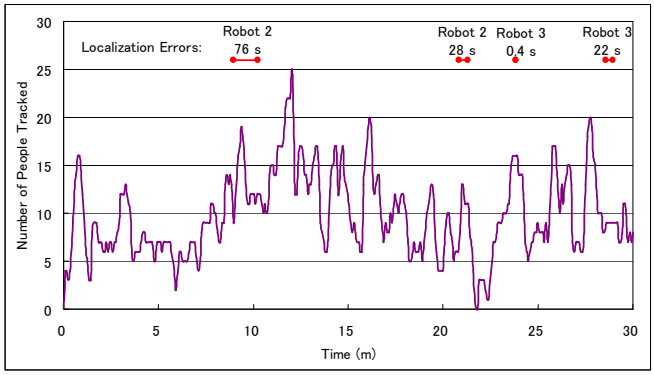

Figure 10. Timing and duration of localization errors for a 30-minute trial with four robots. Localization errors are shown on the top, and the number of people being tracked (excluding the four robots) is shown below. Each error marked represents a tracking error for only one of the four robots.

For the first two errors, the entity tracking Robot 2 disappeared during extended periods of occlusion. Both times, Robot 2 was interacting with customers when it happened, and the system erroneously re-associated the robot with an entity tracking one of the customers. Since the robot and customers did not move during the interaction, the association algorithm was unable to correct the error until the interaction had finished and the customers had walked away.

For the third error, in which Robot 3 was briefly disassociated, the robot had been patrolling in the lower left area of the map, which has poor sensor coverage. Since the tracking system is less accurate in that area, the tracked entity's estimated velocity was incorrect, and the robot was disassociated due to the speed difference. The system recovered from this error quickly, within half a second.

The fourth error was much like the first two, in that the error occurred while the robot was interacting with a customer. The robot and the customer moved away at roughly the same speed for several seconds, but the system recovered when the customer began walking much faster than the robot.

VI. Discussion

Concerning robustness, we have observed the system to perform best when the robots are driving in different motion patterns. When the robots are all stopped for extended periods of time, the system cannot distinguish between them. This ambiguity usually occurs at system startup, and associations are corrected once the robots begin moving.

Also, the system assumes that every robot connected to the system is physically within the tracking area. When this assumption is violated (*e.g.*, a robot goes offline or drives outside of the tracking area) the association algorithm tends to make more errors. We intend to address this issue in future versions of the system.

VII. Conclusions

We have presented a novel localization system for teams of social robots using a network of laser range finders embedded in the environment. We have demonstrated here that the system can perform both localization and theta correction for the robots reliably, even with large numbers of people in the environment, a task which has not yet been demonstrated for multi-robot teams. In the future we plan to continue improving the robustness and reliability of the system.

Acknowledgment

We would like to thank the staff of Universal CityWalk Osaka for their support in this research.

References

[1] W. Burgard, A.B., Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun. The interactive museum tour-guide robot. In *Proc. AAAI Fifteenth National Conference on Artificial Intelligence*, 1998.

[2] B. Mutlu, J. Forlizzi. Robots in Organizations: The Role of Workflow, Social, and Environmental Factors in Human-Robot Interaction, *Proc. 3rd ACM/IEEE International Conference on Human-Robot Interaction (HRI 2008)*, pp. 287-294, 2008.

[3] T. Kanda, T. Hirano, D. Eaton, H. Ishiguro. "Interactive Robots as Social Partners and Peer Tutors for Children: A Field Trial," *Human Computer Interaction*, vol. 19, no. 1-2, pp. 61-84, 2004.

[4] E. Pacchierotti, H. I. Christensen, and P. Jensfelt. Design of an Office-Guide Robot for Social Interaction Studies, *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4965-4970, 2006.

[5] E. A. Sisbot, R. Alami, T. Simeon, K. Dautenhahn, M. Walters, S. Woods, K. L. Koay, and C. Nehaniv. Navigation in the presence of humans. In *Proc. IEEE International Conf. on Humanoid Robots*, pp. 181–188, 2005.

[6] H. Zhao, M. Chiba, R. Shibasaki, X. Shao, J. Cui, H. Zha. SLAM in a Dynamic Large Outdoor Environment using a Laser Scanner, *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nice, France, pp. 1455-1562, 2008.

[7] D. Amarasinghe, G. K. I. Mann and R. G. Gosine. Integrated Laser-Camera Sensor for the Detection and Localization of Landmarks for Robotic Applications, *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nice, France, pp. 4012-4017, 2008.

[8] S. Park, R. Saegusa, and S. Hashimoto. Autonomous navigation of a mobile robot based on passive RFID, *16th IEEE International Conference on Robot & Human Interactive Communication (RO-MAN)*, pp. 218-223, 2007.

[9] A. Saffiotti, M. Broxvall, M. Gritti, K. LeBlanc, R. Lundh, J. Rashid, B.S. Seo, Y.J. Cho. The PEIS-Ecology Project: Vision and Results, *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nice, France, pp.2329-2335, 2008.

[10] M.W.M.G. Dissanayake, P.Newman, S. Clark, H.F. Durrant-Whyte and M. Csorba. A solution to the simultaneous localization and map building (SLAM) problem, Trans. on Robotics and Automation, 17(3), pp. 229-241, 2001.

[11] R. Ouellette, K. Hirasawa, A Comparison of SLAM Implementations for Indoor Mobile Robots. Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2007), pp. 1479-1484, 2007.

[12] 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.

[13] D. Fox, W. Burgard, and S. Thrun. Markov localization for mobile robots in dynamic environments. *Journal of Artificial Intelligence Research*, 2:391–327., 1999

[14] R. Simmons *et al.*, GRACE: An autonomous robot for the AAAI Robot Challenge, *AAAI Magazine*, 24, 2, 2003, 51-72.

[15] M. Montemerlo, S. Thrun and W. Whittaker. Conditional particle filters for simultaneous mobile robot localization and people-tracking. *Proc. IEEE International Conference on Robotics and Automation (ICRA). vol.1 11-15* Washington, DC, USA, May 2002

[16] D.F. Wolf and G. S. Sukhatme. Mobile Robot Simultaneous Localization and Mapping in Dynamic Environments. Autonomous Robots 19, 53–65, 2005

[17] D. Schulz, W. Burgard, and D. Fox. "People tracking with mobile robots using sample-based joint probabilistic data association filters," *International Journal of Robotics Research*, vol. 22, no. 2, 2003.

[18] C. C. Wang, C. Thorpe and S. Thrun. Online simultaneous localization and mapping with detection and tracking of moving objects: Theory and results from a ground vehicle in crowded urban areas. *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, Taipei, Taiwan, September 2003.

[19] T. D. Vu, O. Aycard, N. Appenrodt. Online localization and mapping with moving object tracking, *Proc. IEEE Intelligent Vehicle Symposium*, 2007.

[20] A. Billard, A. Ijspeert, and A. Martinoli. A multi-robot system for adaptive exploration of a fast changing environment: probabilistic modeling and experimental study, *Connection Science*, vol. 11, no. 3-4, pp. 357-377, 2000.

[21] W. Burgard, M. Moors, C. Stachniss, and F. Schneider. Coordinated multi-robot exploration. *IEEE Transactions on Robotics*, vol. 21, no. 3, pp. 376–378, 2005.

[22] D. Pizarro, M. Marron, D. Peon, M. Mazo, J. C. Garcia, M. A. Sotelo, E. Santiso. Robot and Obstacles Localization and Tracking with an External Camera Ring, *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, pp. 516-521, 2008.

[23] T. Kanda, H. Ishiguro, M. Imai, T. Ono. Development and Evaluation of Interactive Humanoid Robots, *Proceedings of the IEEE*, vol. 92, no. 11, pp. 1839-1850, 2004.

[24] 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)*, pp.380-389, 2008.

[25] D. F. Glas, T. Kanda, H. Ishiguro, and N. Hagita. Field Trial for Simultaneous Teleoperation of Multiple Social Robots, *4th ACM/IEEE International Conference on Human-Robot Interaction (HRI 2009)*, pp.149-156, San Diego, CA, 2009.

[26] 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, Mar 2009.

[27] S. Thrun, W. Burgard, and D. Fox. Probabilistic Robotics (MIT Press, 2005).

[28] D. Fox. KLD-Sampling: Adaptive Particle Filters, *Advances in Neural Information Processing Systems (NIPS)* 14, pp. 713–720, MIT Press, 2001.

[29] A. Bruce and G. Gordon. Better motion prediction for people-tracking, in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, New Orleans, LA, USA, 2004.