

Simultaneous People Tracking and Robot Localization in Dynamic Social Spaces

Dylan F. Glas · Yoichi Morales · Takayuki Kanda · Hiroshi Ishiguro · Norihiro Hagita

Received: Oct 2, 2013

Abstract Accurate robot localization and people tracking are necessary for deploying service robots in crowded everyday environments such as shopping malls, and features like product displays change over time, making map-based localization using on-board sensors difficult. We propose the use of an external sensor system to track people together with one or more robots. This approach is more robust to occlusions than on-board sensing and is unaffected by changing map features.

In our system, laser range finders track people and robots in the environment, and odometry data is used to associate each robot with a tracked entity and correct the robot's pose. Techniques are also presented for identifying and recovering from tracking errors. Simulation results show that our system can outperform localization using on-board sensors, both in tracking accuracy and in automatic recovery from errors.

We demonstrate our system's effectiveness in simulation, in a controlled experiment in a real shopping mall environment, and in real human-robot interactions with customers in a busy shopping arcade.

Keywords Robot localization, Social robots, Sensor systems, Dynamic environments

Dylan F. Glas (✉)
ATR-IRC, 2-2-2 Hikaridai, Keihanna Science City, Kyoto, Japan
Tel: +81-774-95-1405, Fax: +81-774-95-1408
e-mail: dylan@atr.jp

Yoichi Morales
ATR-IRC, 2-2-2 Hikaridai, Keihanna Science City, Kyoto, Japan
e-mail: yoichims@atr.jp

Takayuki Kanda
ATR-IRC, 2-2-2 Hikaridai, Keihanna Science City, Kyoto, Japan
e-mail: kanda@atr.jp

Hiroshi Ishiguro
IRL, Osaka University, 1-3 Machikaneyama, Toyonaka, Osaka, Japan
e-mail: ishiguro@sys.es.osaka-u.ac.jp

Norihiro Hagita
ATR-IRC, 2-2-2 Hikaridai, Keihanna Science City, Kyoto, Japan
e-mail: hagita@atr.jp

1 Introduction

The domain of service robots operating within social environments is a growing field of interest in robotics. Many field studies have explored the use of robots in public or commercial spaces, including museums (Wolfram Burgard et al. 1998), hospitals (Mutlu and Forlizzi 2008), schools (Kanda et al. 2004a), and shopping centers (Kanda et al. 2010), as shown in Fig. 1. In many such scenarios, dynamic navigation among pedestrians is necessary in order to provide services, to avoid obstructing people, and to ensure safety.



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

1.1 Localization and human tracking for social robots

For many social robotics applications, high positioning accuracy is a necessity (Glas et al. 2009a). If a robot is incorrectly localized while approaching or facing a customer, it is possible that the customer will not understand that the robot is trying to approach or address them. When giving directions or talking about products in a shop, a robot must point to places or objects to convey meaning. If the pose of the robot is not correctly estimated, the target of the pointing gesture could be completely misinterpreted. Research has also shown that the

nuances of the robot's motion planning are important for psychological reasons (Pacchierotti et al. 2006),(Sisbot et al. 2005). For example, approaching a person from the wrong direction due to a localization error could make that person uncomfortable.

For service robots in everyday environments, the need to track people robustly and accurately is also essential, both for navigation and for interaction. While on-board sensors can be used for tracking people at short range, it is possible to attain more robust, wide-area tracking by using external sensors embedded in the environment. For this purpose, laser range finders (LRFs) can provide high accuracy, and the advent of inexpensive RGBD sensors may reduce the cost of wide-area tracking substantially in the near future.

Such wide-area coverage enables a number of techniques which are important for social robots. Socially-aware path planning for approaching people requires robots to have knowledge of people's trajectories from 6 meters away or further (Satake et al. 2010). Human-factors considerations in motion planning require the robot to know the location of people before it is within line-of-sight, so it does not surprise them by appearing suddenly from behind objects (Sisbot et al. 2005). Other techniques enable robots to anticipate people's future behavior based on the observation of a large environment over time (Kanda et al. 2009).

Thus, although such sensor systems require additional hardware and software infrastructure, they provide considerable value for social robotics applications and can be considered as a necessary framework in many cases. In this work, we make the assumption that an external sensor system for tracking people is available.

1.2 Dynamic social environments

For social robots to function in everyday environments, two main challenges to localization need to be overcome:

First, in busy public spaces, the presence of large numbers of people presents a challenge for traditional map-based localization, as people crowded around the robots can occlude the fixed references that would be used for map-matching.

Second, commercial environments also tend to contain many changing features such as movable furniture and product displays, which will degrade the accuracy of most localization approaches based on map-matching. Fig. 2 shows examples of such changing features from a shopping mall. A localization technique that is robust to such changing maps would be of great value in such environments.

1.3 Proposed solution

We assume that for social robot applications in public or commercial spaces, a high-precision tracking system such as the LRF-based system we use in our field trials will already be embedded in the environment for the purpose of tracking the motion of people. Given this assumption, we propose a robot localization solution which would take advantage of such an infrastructure, requiring no hardware beyond that already in

place for tracking people. The use of sensors fixed in the environment for robot localization should provide a number of benefits.

First, as the sensors themselves are fixed reference points, they should enable more precise localization than that attainable by on-board sensing.

Second, while any sensors, onboard or offboard, are susceptible to occlusions in crowded environments, the use of multiple sensors positioned around the environment should result in fewer occlusions than use of a single on-board sensor.

Third, as this localization technique is not based on map-matching, it will provide consistent accuracy in environments with changing features.

In this paper we describe our proposed system and evaluate its performance through simulations and field trials. We show that its accuracy is superior to that of a map-matching approach, particularly when the environment has changing features, and we demonstrate its real-world performance in a field trial at a busy shopping arcade.



Fig. 2 Changing product displays. Photos on the left and right show how features in three areas of the shopping mall differ on different days.

2 Related Work

2.1 Map-based localization

2.1.1 Online SLAM

In the robot navigation community, much effort has been devoted to the problem of simultaneous localization and mapping (SLAM) using onboard sensors (Dissanayake et al. 2001),(Ouellette and Hirasawa 2007). These techniques serve the dual purpose of map-building and robot localization. Some research additionally addresses the tracking of people in an environment, e.g. (Montemerlo et al. 2002),(Wolf and Sukhatme 2005),(Dirk Schulz et al. 2003).

Related research using road vehicles has focused on combining SLAM with the detection of moving objects in urban environments, such as (Wang et al. 2003) and (Vu et al. 2007).

Collaborative SLAM with multiple robots is also an active field of research (Billard et al. 1999), (W. Burgard et al. 2005), 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.

2.1.2 Offline map-matching

While SLAM is a major topic of research in robotics, the full process of building maps online is often unnecessary. For robots operating in relatively static, known environments, a map-matching approach based on a previously-constructed map, such as Monte-Carlo Localization (Thrun et al. 2001), is sufficient, and it requires less computational overhead than solving the general SLAM problem.

Localization based on previously-generated maps has been used quite often in scenarios such as Robocup (Köse and Akın 2007), the DARPA Urban Challenge (Urmson et al. 2008), and for security robots (Takahashi 2007), expo guide robots (Siegwart et al. 2003), and shopping guide robots (Gross et al. 2009).

As map-based approaches rely on stability of the environment, they are usually not suitable for environments in which the features are expected to change frequently. However, some techniques have been proposed for handling such dynamic updates.

For example, the technique presented in (Lee and Chung 2010) uses recognizable features for localization and uses observations of unmodeled features to update its internal map. However, this is an offline technique, in which a map-updating process is conducted offline based on an interpolation of the robot's path. At the time while it is moving through areas with unmodeled features, the robot has no information to help correct its localization. In addition, their technique requires at least 30% of the robot's observations to correspond with the known map, whereas our proposed solution can work even when 100% of the features have changed.

2.2 Localization using external infrastructure

At the same time, other researchers are developing highly accurate and robust navigation and mapping techniques for robots in exploratory and military applications. Some of these techniques are based on external references such as GPS (Zhao et al. 2008) or fixed landmarks (Amarasinghe et al. 2008), and others are based on visual or RFID tagging of the robot or environment (Park et al. 2007), or integration with intelligent environments (Saffiotti et al. 2008).

While these approaches can localize a robot with high accuracy, their typical drawback is that they require the installation of external infrastructure solely for the purpose of supporting localization.

However, we already assume the existence of a sensor system embedded in the environment for human tracking, and this infrastructure can be used for localization with no additional cost. A variety of such tracking systems have been developed around the world (Fod et al. 2002; Bennewitz et al. 2005; D. Schulz et al. 2003; Cui et al. 2007)

Our approach to robot localization is to use such fixed external sensors rather than the on-board sensors of the robots. Our implementation uses laser range finders, but visual or RGBD cameras could also be used, without loss of generality. As many different tracking systems exist, one of our design goals is to preserve the abstraction barrier separating the implementation of the underlying people-tracking infrastructure from the robot localization technique.

The work of Pizarro et al. (Pizarro et al. 2008) is similar to our approach, using a ring of cameras and an extended particle filter technique for robot localization. However, the technique presented in that work requires knowledge of the robot's initial position. Our approach uses information which is independent of position and orientation to match between odometry and laser data, so it can be used with no knowledge of the robot's initial position and is capable of recovering from the "kidnapping" problem.

3 Tracking Infrastructure

The algorithm to be presented in the paper does not depend on a particular tracking infrastructure, as long as some basic requirements are met.

3.1 Minimum Requirements

The minimum infrastructure required for our technique is a system which can anonymously report the (x, y) positions of entities (humans or robots) in the environment.

As our system is proposed for use with humanoid robots, we assume that the system cannot reliably distinguish between humans and robots by shape alone. This assumption stems from experiences with our field systems, where we have found that it can be quite difficult to differentiate between robots and humans based on range data alone because variations in reflectivity, clothing, and accessories such as backpacks and handbags produce a large variation in the shape profiles of people.

3.2 Tracking systems used in this research

We used two configurations of laser range finders in our studies. The number and type of sensors used differed between the configurations, but the tracking algorithm and processing of data were the same. In both cases, sensors were placed around the perimeter of the environment to minimize the likelihood of occlusions.

3.2.1 Hardwired configuration

Our first hardware configuration included six SICK

LMS-200 laser range finders with long-range sensitivity. They were usually 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 every 26 ms.

The LRF's 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.

As the sensors in this configuration have a high power draw, they were hardwired into the environment. The sensors were connected directly to power supplies and a central data acquisition PC, which then streamed all sensor data to the tracking server (a separate PC).

3.2.2 Wireless configuration

The other configuration was a portable, wireless system using lighter-weight Hokuyo UTM-30LX Top-URG laser range finders with a nominal detection range of 30 m. Each sensor was mounted at the top of a pole at a height of 83 cm. For this configuration, the sensors were set to scan an angular area of 180° at a resolution of 0.25° , providing readings every 25 ms.

The data from each of these sensors was acquired using a portable Asus Eee PC 901 netbook mounted inside the base of the pole. Both the PC and the sensor were powered by batteries stored inside the base, with a maximum continuous data capture duration of about 10 hours.

In this configuration, the capture PC's streamed data to the tracking server (a separate PC) over a wireless LAN.

3.2.3 Tracking Algorithm

For the experiments in this paper, we used a tracking algorithm using individual particle filters to track multiple entities. Details of our detection and tracking algorithms are presented in (Glas et al. 2009b). To briefly summarize the tracking algorithm, foreground scan segments are identified for each sensor using background subtraction, and these segments are used to define a likelihood model for the locations of tracked entities. The likelihood models for all sensors are combined, and rules are established for creating individual particle filters for each new human (or robot) detection and deleting them when the entities are no longer being observed. The particle filters themselves model the entity's position, orientation, and linear velocity (x, y, θ, v) , and the update model projects each particle by projecting its linear motion and adding normally-distributed noise in all four variables.

A similar approach is taken in (Brščić et al. 2013), where head detections from several ceiling-mounted 3D depth sensors are combined to track people using individual particle filters. Although the results in this paper have been evaluated assuming an LRF setup, in practice we often use the same

localization technique with both tracking systems.

4 Proposed Technique

The localization technique we propose here consists of two steps executed iteratively. In the first step, association is performed between the odometric trajectories of one or more robots and tracked trajectories of anonymous tracked entities based on similarity of their motion. Then in the second step, pose corrections are applied to the associated robots.

In this work, the basic task of estimating track continuity is conducted by the tracking system. This is not a trivial problem, and we assume that any tracking system will make errors regarding track continuity, such as merging or splitting tracks incorrectly. As we will show later, by dynamically managing track associations, our proposed technique is able to recover quickly and function effectively despite errors which occur in the tracking system.

4.1 Related techniques

The fundamental problem to be addressed is the assignment of a known robot's odometric track to a track observed by the environmental sensors, which shares some similarities with the general data association problem of multiple target tracking systems, e.g. for radar. The classical problem is typically addressed by techniques such as Multiple Hypothesis Tracking (MHT) and its variants (Reid 1979; Blackman 2004).

However, there is a fundamental difference with the classical tracking problem. MHT systems are typically used for associating noisy observations in absolute spatial coordinates to previously-observed tracks in that same coordinate system. However, the problem we face is the association of *extrinsic* measurements from the tracking system with *intrinsic* measurements of odometry reported by the robot.

As these observations are not in the same coordinate system, similarity between tracks cannot be directly measured, e.g. by Cartesian distance. A new technique is needed to enable track association between incompatible coordinate systems.

4.2 Problem Formulation

To associate odometric trajectories of the robots with tracked trajectories of entities detected by the tracking system, we examine the recent history of data from each source, represented in the form of a trajectory. We define a trajectory T to be comprised of a sequence of n points, $T = [p_1, p_2, \dots, p_n]^T$. Each point $p_i = [x_i \ y_i \ t_i]^T$, represents the entity's (x, y) position at time t_i . For tracked trajectories, we obtain p_i directly from the observed positions for these points, and for odometric trajectories, we generate p_i by integrating the v and ω values reported by the robot based on odometry. Note that tracking systems vary in their capabilities, so our algorithm does not assume that orientation is explicitly observable in data provided by the tracking system.

Let N_R be the number of robots to be tracked, and let N_T be the number of trajectories observed, where $N_T \geq N_R$. We are

given a list of robot trajectories $R_i \mid i \in (1 \dots N_R)$, computed from odometry and defined in robot-relative coordinates, and a list of observed trajectories $T_j \mid j \in (1 \dots N_T)$, defined in the global coordinate frame of the sensor system. We wish to identify a unique mapping from each robot's odometric trajectory R_i to one observed trajectory T_j .

4.3 Evaluating Trajectory Similarity

To estimate the likelihoods of correspondences between a robot's odometric trajectory and observed trajectories, we define a similarity function to compare trajectory shapes.

4.3.1 Data preprocessing

We assume the trajectories are recorded in the system at regular time intervals (we chose 200 ms). If this is not the case, recorded trajectories can be resampled to regular time intervals. Humans can be detected at any time, so we compare only the overlapping time range t_{valid} during which the trajectories being compared, R_i and T_j , both contain valid data.

If t_{valid} is too short, there is not enough shape information to distinguish trajectories, so comparisons are only performed if t_{valid} is at least t_{min} seconds long (we used 5 s), and the overlapping data is truncated after t_{max} seconds (we used 15 s) to simplify computations.

4.3.2 Measuring trajectory similarity

To compare the motion reported by the robot with the motion observed by the sensors, we define a metric quantifying spatiotemporal similarity between trajectories. Only pose-invariant features are candidates, as they must be comparable between the sensor and robot coordinate frames. Candidates for such metrics include velocity, acceleration, trajectory curvature, or other features. We chose to use the geometric similarity between the trajectory shapes, because the geometry implicitly incorporates all of the above features.

To compare the shapes of two trajectories with known correspondences (based on time of detection), we must determine the optimal rigid 2D transformation to minimize the error between them. We thus compute the homogeneous transformation matrix \mathbf{A} which minimizes ϵ , the squared Euclidean error between corresponding trajectory points:

$$\epsilon = \|\mathbf{A}\vec{R}_i - \vec{T}_j\|^2.$$

We define \mathbf{A} as a function of parameter vector β as follows:

$$\beta = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \quad \mathbf{A}(\beta) = \begin{bmatrix} \cos\theta & -\sin\theta & x \\ \sin\theta & \cos\theta & y \\ 0 & 0 & 1 \end{bmatrix}$$

The optimal values of β which minimize ϵ are then computed. The least-squares solution can be obtained through common iterative techniques such as the Levenberg-Marquardt algorithm, or through singular-value decomposition using the Kabsch algorithm (Bevilacqua et al. 2006).

The minimum value of ϵ can then be used as a metric characterizing the difference between the trajectory shapes.

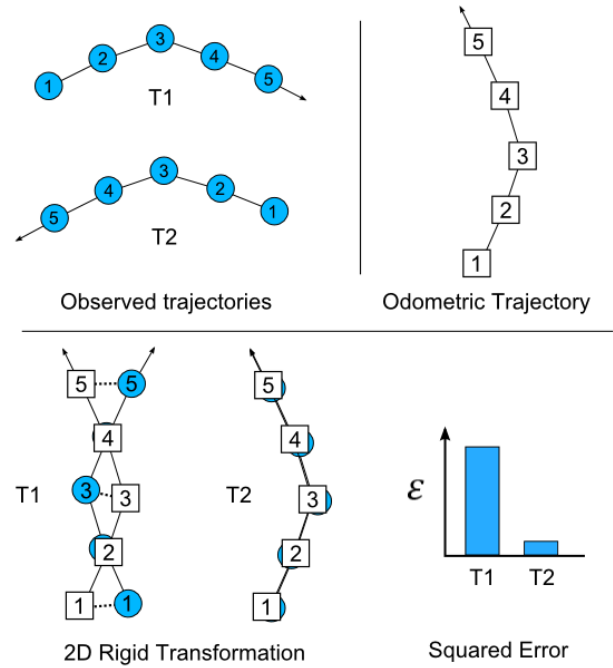


Fig. 3 Trajectory shape comparison. Trajectories T1 and T2 are observed by the sensor system and compared with an odometric trajectory from the robot. A 2D rigid transformation is applied to obtain the best fit between observations which correspond in time, and the squared error, ϵ , is computed for each possible match.

Since the number of points being compared can vary depending on the length of t_{valid} , the error is normalized by the number of points compared, such that the final metric used is $\hat{\epsilon} = \epsilon/n$. Figure 3 illustrates the trajectory comparison process.

4.4 Track Association

Once the similarity metric $\hat{\epsilon}$ can be obtained for each trajectory pair, the next step is to make decisions about track assignments. For this step, two principal requirements must be considered.

4.4.1 Requirements

a) Reliably identify matching tracks

Robots will often have similar trajectories to each other – robots that interact with people might stop for long periods of time to have conversations, and even moving robots may be indistinguishable from each other based on trajectory shape, e.g. if they are the same build of robot with a fixed maximum speed, and they are moving in straight lines.

Figure 4 shows an illustration of this phenomenon. Robots are distinguishable from each other only at certain times, so it is risky to make new track assignments during the time periods shaded in gray in the figure.

For this reason, the policy of our approach is to create robot-track assignments at times when correspondences can be identified with high confidence, and then to maintain those assignments as long as the correspondences remain valid.

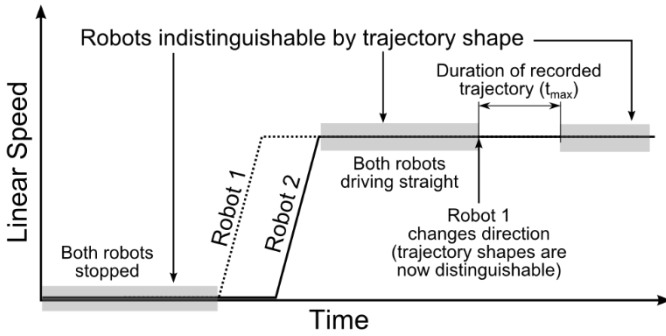


Fig. 4 During some periods of time, robot trajectories will be indistinguishable from each other, such as when robots are stopped or driving in a straight line at their maximum speed (gray areas in the figure). Assignments can only be made at times when robot motion is unique. Note that although for simplicity this graph only shows instantaneous speed, the trajectory match is actually performed using a Cartesian match of trajectory shapes.

b) *Recover quickly from track continuity errors*

Stable tracking is a challenging task, and we assume the tracking system will occasionally encounter tracking errors. For example, two entities moving near each other could be mistaken for each other and their ID’s could become switched. In these cases, the track association of the robot must be updated as quickly as possible to the correct tracked entity. For this reason, our policy is to disassociate a robot as soon as possible if there is any evidence that it may not be correctly assigned. At every time step, we evaluate three metrics to identify incorrect associations: a robot is disassociated if track similarity $\hat{\epsilon}$ goes below a threshold, if its instantaneous speed v differs from its tracked entity by more than a threshold, or if its projected Cartesian position differs from that of its tracked entity by more than a threshold distance.

4.4.2 *Assignment Algorithm*

a) *State Representation*

To satisfy these two requirements, we use a state model to represent a robot’s association status. Once a robot is correctly associated with a tracked entity with high confidence, that assignment will be maintained until there is evidence that the association is no longer correct. In this way associations remain stable even during times when they cannot be distinguished by trajectory shape. We reflect this information by assigning each robot an association state $\mathbb{S}_{R_i}(t)$ which can either *associated* or *unassociated*, as shown in Fig. 5.

At each update of the algorithm, the existing assignments of all *associated* robots are evaluated to determine whether they are still valid. The trajectories of all *unassociated* robots are then compared with those of the tracked entities, and new associations are created if possible. The complete procedure for the association algorithm is presented in Table 1.

b) *Disassociation*

At each cycle of the association algorithm, the first step is to identify any incorrect associations and disassociate them.

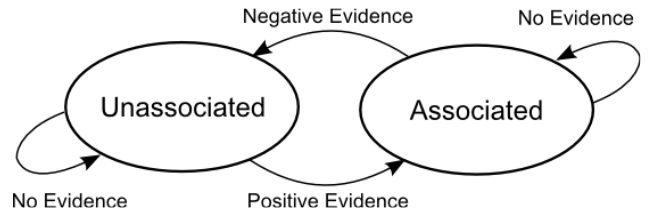


Fig. 5 State transition model for robot-entity associations.

Incorrect associations can occur due to track continuity errors from the tracking system or incorrect mappings from the association algorithm.

The algorithm will transition a robot from *associated* to *unassociated* if there is any negative evidence for its current association, specifically if any of the following three conditions are satisfied:

- The tracked entity to which the robot is associated is no longer detected
- The robot’s motion is too inconsistent with the motion of the tracked entity
- This robot’s associated entity is the best-match entity for a different, unassociated robot

After applying these rules for disassociation, the algorithm proceeds to associate unassociated robots when possible.

c) *Association*

We next evaluate which robots should be transitioned from *unassociated* to *associated*. A cost matrix \mathcal{M} is computed by evaluating $\hat{\epsilon}$ for every trajectory pair candidate (R_i, T_j) considering all *unassociated* robots and all tracked entities.

A global nearest neighbor matching is then performed to dynamically assign association mappings to whichever set of robot-trajectory mappings minimizes the sum of $\hat{\epsilon}$. This is the classic 2D assignment problem, which can be solved using techniques such as the Simplex method or the Hungarian method (Kuhn-Munkres algorithm). Note that although strictly speaking, the Hungarian algorithm is used to assign n jobs to n workers, variants exist which can assign n jobs to m workers, where $m > n$. We used Kevin L. Stern’s implementation¹ for assignment, although many similar implementations exist.

If the optimal association map assigns a new robot to a tracked entity previously assigned to a different robot, both robots are disassociated and the process is repeated to find the best set of assignments.

4.4.3 *Threshold value determination*

Several threshold values are used in this algorithm, and we have not developed a formal method for determining these values. Several factors affect the tuning of the parameters.

Quality of data from the tracking system can vary based on geometry of the sensors, complexity and crowding of the environment, and reflectivity of the robot(s) and other objects.

¹

<http://software-and-algorithms.blogspot.jp/2012/09/the-hungarian-algorithm-for-assignment.html>

In cases where tracking is less reliable, e.g. the entity associated to the robot disappears frequently, factors such as t_{min} , $dist_threshold$, and $speed_threshold$ can be reduced, yielding jumpier tracking but lowering the risk that tracking glitches will cause long-lasting misassociations.

Odometry error can also vary based on the environment, e.g. smooth floors vs carpeted areas. If there is a great deal of wheel slip and odometry is less reliable, it could be useful to increase $error_threshold$ in order to allow association based on looser matches between odometric and tracked trajectories.

4.4.4 Computational load

We measured the amount of time required to execute the trajectory pair comparisons and association algorithm, executed in Java on an Intel i7 processor.

For this comparison, we generated random trajectories corresponding to the robots and pedestrians, using 25 data points for each trajectory and adding random noise to each point for the sensor data. For various settings of number of robots, number of humans, and number of currently-associated robots, we averaged the computation time of the association algorithm across 10 trials using different trajectories.

The computational load is strongly dependent upon the number of robots which are currently associated – for h humans, a associated robots, and u unassociated robots, the number of trajectory comparisons for the associated robots is only a , since it is only necessary to confirm that their tracks are still valid. For the unassociated robots, each robot must be compared with the tracks from all humans and all robots, or $u(h + u + a)$ comparisons. Thus, as more robots become successfully associated, the computational load decreases significantly.

In our simulation, we considered three scenarios. The first was a typical case from one of our field deployments, with four robots deployed in a space along with an average of 40 people. We also simulated heavier loads of 10 robots with 100 people and 20 robots with 200 people. For reference, the largest deployment of our tracking system has tracked a maximum of about 200 people at once.

Results of our benchmark test are shown in Fig. 6. For all cases, the maximum computation time required for full association of all robots was well within our chosen update interval of 200 ms. For the case where $u = 4$, $h = 40$, the total time was just over 7 ms.

Closer inspection shows that most of the computation time was used by trajectory comparison. Assignment determination by the Hungarian Algorithm required an average of 0.04 ms for the 4-robot scenario, 0.18 ms for the 10-robot scenario, and 0.68 ms for the 20-robot scenario, and was mostly independent of the number of robots to be associated.

In practice, the majority of the robots will usually be associated most of the time – in both the simulation and field studies presented later in this paper, we will see that the robots were associated over 99% of the time. Thus, we can expect that even for large teams of robots, computation time will typically

Table 1. Association update algorithm

| ALGORITHM <i>update-associations</i> | |
|---|--|
| INPUTS: $R_i \mid i \in (1 \dots N_R), T_j \mid j \in (1 \dots N_T)$ | |
| $\forall R_i$, update $\mathbf{x}^{R_i}, \dot{\mathbf{x}}^{R_i}$ of R_i from odometry | |
| $\forall T_j$, update $\mathbf{x}^{T_j}, \dot{\mathbf{x}}^{T_j}$ of T_j from tracking system | |
| for $i : 1 \sim N_R$ % Disassociate invalid matches | |
| if $\mathbb{S}_{R_i} == associated$ | |
| Compute $\hat{\epsilon}$ for $[R_i, T_j]$ | |
| Compare position and velocity of R_i with associated T_j . | |
| if ($\hat{\epsilon} > error_threshold$ | |
| OR $\ \mathbf{x}^{R_i} - \mathbf{x}^{T_j}\ > dist_threshold$ | |
| OR $\ \dot{\mathbf{x}}^{R_i} - \dot{\mathbf{x}}^{T_j}\ > speed_threshold$) | |
| $\mathbb{S}_{R_i} := unassociated$ | |
| end if | |
| end if | |
| end for | |
| for $i : 1 \sim N_R$ % Compute cost matrix \mathcal{M} | |
| if $\mathbb{S}_{R_i} == unassociated$ | |
| for $j : 1 \sim N_T$ | |
| Compute $\hat{\epsilon}$ for $[R_i, T_j]$ | |
| if $\hat{\epsilon} \leq error_threshold$ | |
| $\mathcal{M}(i, j) := \hat{\epsilon}$ | |
| end if | |
| end for | |
| end if | |
| end for | |
| LABEL 1: % Create associations | |
| for $(R_i, T_j) \ni \mathcal{M}(i, j)$ is defined | |
| Compute assignment map \mathbb{A} , e.g. via Hungarian Algorithm | |
| end for | |
| while \exists entity T_j associated to more than one robot | |
| for $R_i \ni R_i$ associated to T_j | |
| if $\mathbb{S}_{R_i} == associated$ | |
| $\mathbb{S}_{R_i} := unassociated$ | |
| Compute $\hat{\epsilon}$ for $[R_i, T_j]$ | |
| if $\hat{\epsilon} \leq error_threshold$ | |
| $\mathcal{M}(i, j) := \hat{\epsilon}$ | |
| end if | |
| end if | |
| end for | |
| Go to LABEL 1 | |
| end while | |
| for (R_i, T_j) in \mathbb{A} | |
| Assign $R_i \rightarrow T_j$ | |
| $\mathbb{S}_{R_i} := associated$ | |
| end for | |

stay towards the low end of each graph. In practice, our localization server usually runs on the same PC as other

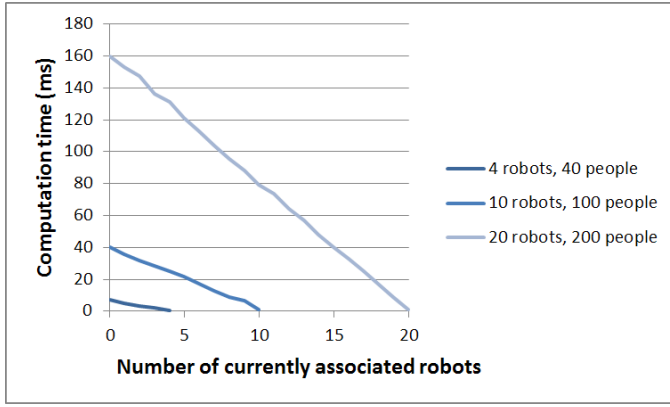


Fig. 6 Computation time for the association algorithm.

software such as the tracking system, and the extra computational load due to the localization system is quite small.

4.5 Applying Robot Pose Corrections

Once associations have been determined, position and heading corrections must be applied to all robots which have been successfully associated to tracked entities.

As our sensor network can perform tracking with very high precision (a mean error of 4.6 cm was reported in (Glas et al. 2009b), although the accuracy varies based on geometry of the sensor configuration), we chose to directly apply the detected (x, y) positions to the robots, rather than using an estimation filter incorporating an odometry model. Since orientation is not observable from the tracking system, correction of the robot’s heading is computed separately using a 1-dimensional Kalman filter.

4.5.1 Position correction

For each robot, the position data from the laser tracking system is used to update the robot’s (x, y) position.

If the robot is moving, we set its position directly to the latest observed position of its associated entity. In the case that a robot is stopped, which can be accurately determined using wheel encoders, we can obtain a more stable position estimate by averaging its position estimates over time. If odometry indicates that a robot is stopped, we add the latest observed position to a window filter for averaging.

4.5.2 Heading correction

Although the (x, y) position estimates sent to the robots have some error, that error will be no larger than that of the tracking system itself. Heading errors are more critical, however, as they will more strongly cause odometric position estimates to diverge from the robot’s true position over time. A mechanism is thus needed to correct orientation errors.

Stated more precisely, the purpose of this step is to correct the angular error between the robot’s on-board estimate of the world coordinate system and the actual world coordinate system. For a differential-drive robot, this can be seen as

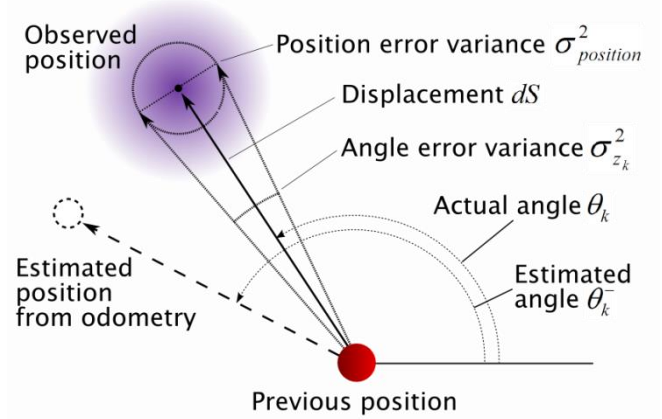


Fig. 7 Dynamic model showing estimated pose, observed pose, and covariance of measurement error.

correcting the robot’s heading. For an omnidirectional robot, this can be interpreted as aligning its estimated direction of motion with its externally-observed direction of motion.

As noted earlier, we do not assume that the tracking system can directly observe the robot’s orientation, so we use a one-dimensional Kalman filter to estimate this variable based on observed displacements.

Kalman filtering is a classic 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 1-3 show the form which most clearly illustrates our approach.

$$K_k = \frac{\sigma_{z_k}^2}{\sigma_{z_k}^2 + \sigma_{\theta_k^-}^2} \tag{1}$$

$$\hat{\theta}_k = \theta_k^- + K_k (z_k - \theta_k^-) \tag{2}$$

$$\sigma_{\hat{\theta}_k}^2 = \sigma_{\theta_k^-}^2 + K_k (\sigma_{\theta_k^-}^2) + \sigma_{process}^2 \tag{3}$$

Our aim is to recursively approximate the actual motion direction θ_k at time step k with an estimate $\hat{\theta}_k$, given angle observation z_k with measurement variance $\sigma_{z_k}^2$ and previous motion direction estimate θ_k^- .

The variance $\sigma_{\hat{\theta}_k}^2$ of the motion direction estimate is updated at each step, and it is assumed that this increases over time due to a process noise with variance $\sigma_{process}^2$. The term K_k in Eq. 1 represents the Kalman gain, which is used in calculating the new angle estimate (Eq. 2) and its variance (Eq. 3). As each new observation is recorded, these three equations are evaluated recursively, using the previous step’s state estimate and variance.

The state being estimated in this case is the robot’s direction of motion, based on comparing the robot’s estimate of its motion from odometry with observations of the robot’s direction of motion from the tracking system, as illustrated in Fig. 7.

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

Prediction phase: Beginning with the robot's previous position $(x_{k-1}^{odo}, y_{k-1}^{odo})$ (which has already been corrected to coincide with the observed position from the previous time step), we project the robot's position based on the odometric translation and rotation reported by the robot to obtain a new position (x_k^{odo}, y_k^{odo}) . The predicted angle θ_k^- is calculated as follows:

$$\theta_k^- = \tan^{-1} \left(\frac{y_k^{odo} - y_{k-1}^{odo}}{x_k^{odo} - x_{k-1}^{odo}} \right)$$

We assume that this motion is corrupted with a small process noise $\sigma_{process}^2$ at each step (we used an arbitrary small value of 0.0001 radian).

Update phase: An observed value for the robot's motion direction is then obtained from the position data from the tracking system. The angle observation z_k is obtained by comparing the position of the tracked entity's position from the sensor network (SN) at step k with its previous position at step $k-1$ as follows:

$$z_k = \tan^{-1} \left(\frac{y_k^{SN} - y_{k-1}^{SN}}{x_k^{SN} - x_{k-1}^{SN}} \right)$$

The position observation is noisy, with a measurement uncertainty $\sigma_{position}^2$. Using a small-angle approximation, we can say that for displacement dS , the variance of the angle error $\sigma_{z_k}^2 \approx \sigma_{position}^2 / dS$.

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.

4.6 Synchronization

Since the tracking infrastructure is connected with the robot over a wireless network, network delay can sometimes be a problem. We have developed a few techniques to minimize the impact of network delay.

First, communication between the robot and the localization server is asynchronous. The robot continues to send updates of its position whether or not it receives any corrections in return. If packets from the robot are delayed, the server will usually receive a burst of several delayed readings. It can then perform its calculations using the most recent update rather than using old data.

Second, the corrections to the robot's orientation are sent as the angular difference between the robot's reported orientation and its estimated orientation, rather than sending the absolute angle. If the robot receives an absolute angle correction with some delay, when it has already rotated to a different angle, the resulting pose can be off by a significant amount, and this error will affect all future pose estimates. By sending the angular difference rather than the absolute angle, we avoid this problem, as the robot can simply apply the correction to whatever its current pose is.

Of course, application of delayed (x, y) position data can

cause an error if the robot is moving quickly, but the offset is just a small constant value, and the long-term consequences are not nearly as serious as they are with heading corrections.

Finally, we often use this system in conjunction with a teleoperation system, by which robot positions can be manually reset remotely.

Care must be taken that manual position corrections to the robot are not immediately overwritten by data from the localization server, which is based on the previously reported position. The nature of asynchronous updates can sometimes 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 address this problem, whenever the robot's position is set to a new location by an external source, 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.

5 Comparison with Map-Matching in Simulation

To evaluate the effectiveness of the proposed technique, we performed simulated trials comparing the proposed method with localization by map-matching, which we consider to be an approach which would typically be used for robot localization in public spaces, e.g. (Thrun et al. 1998).

5.1 Performance Comparison

The simulations were performed using the Player/Stage system (Gerkey et al. 2003). We simulated a robot driving through a virtual environment containing 10 simulated pedestrians, and we recorded odometry data for the robot as well as range data from the robot's on-board LRF and 12 LRF's fixed in the environment.

For each trial, two trajectories were recorded for the robot. First, the robot was manually driven once around the environment to gather data for building a scan map. Next, the robot autonomously wandered around the environment avoiding obstacles for 20 minutes to generate data for performance evaluation. In total, 10 trials were conducted with the simulation, each starting from a different location in the virtual world. Every trial included a 1-2 minute mapping run and a 20-minute evaluation run.

To evaluate our proposed technique, we played back the odometry and laser data logged from the evaluation runs, processing data from the environmental LRF's and the robot's odometry with the same tracking server developed for use with our robots in the field. The resulting corrections were applied to the simulated robot, and the robot's corrected positions were recorded for comparison with ground truth from the simulation.

To evaluate the performance of a map-matching approach, we built a scan map based on the mapping run for each trial, then used the robot's odometry and LRF data from the evaluation run as inputs to a simulated robot that localized itself using a map-matching technique. These results were also recorded for comparison with ground truth.

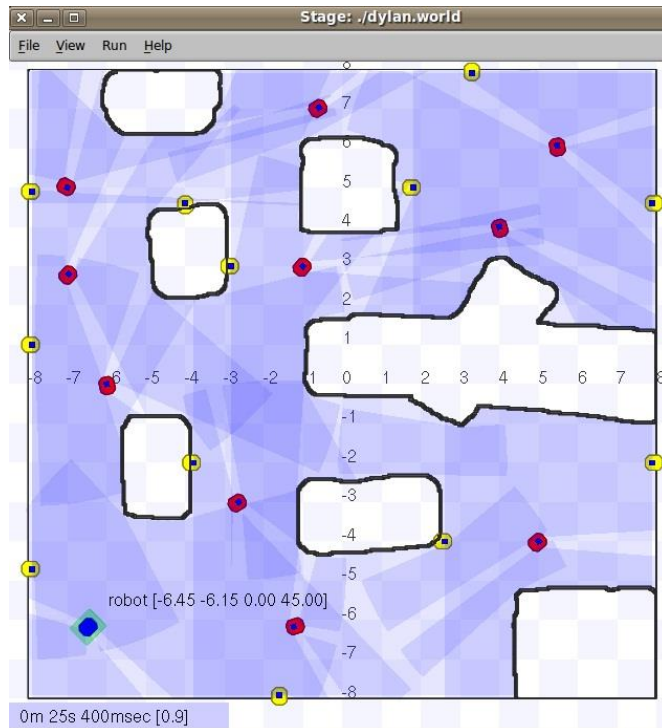


Fig. 8 Screenshot of stage simulation, including robot (blue, in lower left), 10 humans (red, placed around the environment) and 12 sensors (yellow, placed against walls). The shaded areas represent simulated laser scan data from the sensors and robot.

5.1.1 Simulation settings

For the simulation, we used Player 3.1.0 and Stage 3.2.1 running on an Ubuntu Linux system. The world used in these simulations was the “cave” world included with the Stage distribution. One robot and 12 sensors were simulated in this world, as depicted in Fig. 8, and an example trajectory from one of the trials is shown in Fig. 9. Sensors were arranged in an attempt to provide coverage by at least two sensors over most of the open area of the world. Their positions were fixed and not moved between simulations.

The robot was modeled as a MobileRobots Pioneer 2-DX differential-drive robot, and both the robot’s on-board sensor and the environmental sensors were modeled as SICK LMS-200 laser range finders.

To simulate real sensor data, noise was added to both laser and odometry data, with a standard deviation of 0.05 m for laser data and 0.03 m/s for the left and right wheel speeds, which are the default noise values provided for the Pioneer 2-DX model in the Stage simulator.

The pedestrians were also simulated using the Pioneer 2-DX model, as its width of 0.5 m is comparable to that of a person. Both the pedestrians and the robot were configured to move according to the same “wander” model, at a speed of 0.4 m/s. Although human movement in many real environments would be faster than a robot and exhibit different dynamics, we chose to use the same motion behaviors for the humans and robot in order to provide the most challenging test of our system’s ability to identify the robot among many similar trajectories.



Fig. 9 Example of a 20-minute robot “wandering” trajectory used for evaluation, shown on a map of walls and sensors. Humans were continually moving and so are not shown in this diagram.

5.1.2 Evaluation

To measure the performance of our proposed technique, we ran our localization and tracking systems using the robot odometry and environmental LRF data logged from the simulation, and position corrections were sent to a simulated robot and logged.

For the map-matching comparison, we aimed to use commonly available tools and techniques to evaluate the performance of a “typical” map-based localization technique. The 3DTK package available on the openSLAM.org website (3DTK) was used as a representative example of software that is commonly used for map generation.

For each trial, we generated a grid map offline using scan data and odometry data from the mapping run of the simulated robot. To generate localization estimates, we then replayed laser scan data and odometry data from the evaluation run offline, and we used a particle filter to estimate the robot’s position using a ray-tracing approach to estimate particle likelihood. Specific details of our map-building and localization techniques are presented in the Appendix.

In both cases, the resulting robot positions were compared with ground truth to determine absolute localization error at each time step.

5.2 Results and Discussion

The total data set consisted of 82870 data points, representing 10 trials lasting 20 minutes each. We performed two analyses of the data: one comparing the overall error

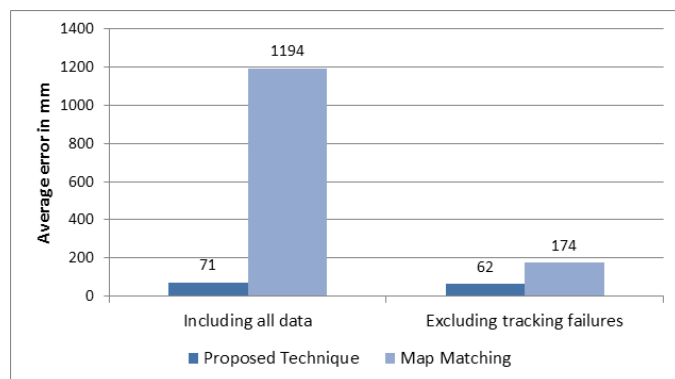


Fig. 10 Comparison of accuracy between the proposed localization method and a map-based localization technique. Extended tracking failures caused the overall average error to be much higher for the map-matching system. Even excluding these tracking failures, our proposed system still outperformed the map-matching approach in simulation.

observed between conditions, and a second analysis considering the effects of large errors due to tracking failures. These results are shown in Fig. 9 and in Table 2.

5.2.1 Localization Error

We compared the mean error for each of the ten trials between the two conditions: *proposed-technique* and *map-matching*. In our analysis, we first checked whether the data followed a normal distribution, using the Shapiro-Wilk test. Analysis for data from each condition showed significance ($p=.028$ for *proposed-technique*, $p<.001$ for *map-matching*), so the null hypothesis that the data are normally distributed was rejected, and we concluded that they do not follow a normal distribution. We compared the errors using a Wilcoxon signed-rank test, which revealed a significant difference between conditions ($p=.005$).

These results, shown in Fig. 10 (left) show that overall, the proposed method significantly outperformed map-based localization using on-board sensors.

5.2.2 Tracking failures

In the map-matching condition, the robot sometimes got lost, unable to match its laser scans to the environment map. In such cases, localization error can grow without bound. Although some recovery solutions have been proposed for the “kidnapped robot problem,” scan-matching alone is not intrinsically able to recover from such failures.

A similar phenomenon can also occur using our proposed technique. If a robot is associated to the wrong tracked entity, its localization error can suddenly become very large.

The mean localization error thus represents a combination of a steady-state level of error while the system is working correctly (correct map-matching or correct association), and a potentially large contribution of error when a “tracking failure” occurs in either map-matching or the proposed method.

It is not trivial to identify when a tracking failure has actually happened, so we chose to consider any error larger than a threshold to be a “tracking failure” in our analysis. We chose a

Table 2. Results from comparison in simulation

| Measurement | Map-matching | Proposed System |
|---|--------------|-----------------|
| Mean Error (s.d.) in mm | 1194 (3964) | 71 (129) |
| Number of tracking failures | 7 | 12 |
| Longest tracking failure duration | 841.1s | 15.9s |
| Mean tracking failure duration | 126.3s | 6.4s |
| % time in tracking failure state | 7.55% | 0.65% |
| Mean error in mm during tracking failure | 13690 | 1430 |
| Mean Error (s.d.) in mm Excluding tracking failures | 174 (107) | 62 (64) |

threshold of 1 m, although very similar results can be observed for smaller or larger thresholds, as the purpose is simply to separate small errors observed in normal operation (e.g. 10 cm) from large errors due to tracking failures (e.g. 5 m).

To examine the average error during periods of successful localization, we removed all tracking failures from the dataset and compared the mean localization error for each of the ten trials between the two conditions, the results of which are shown in Fig. 9 (right).

To check whether the data followed a normal distribution, we again ran the Shapiro-Wilk test. Data from each condition did not fall into significance ($p=.057$, $p=.085$), thus we concluded that they follow a normal distribution. Then, we conducted a repeated-measures ANOVA, which showed a significant difference between conditions ($F(1,9)=113.911$, $p<.001$, $\eta^2=.927$). These results indicate that although a majority of the localization error in the map-matching system can be attributed to tracking failures, the proposed system still significantly outperforms the map-matching system even when those errors are ignored.

Furthermore, an important advantage of our technique is its ability to quickly recover from tracking failures. As Table 2 shows, although our algorithm encountered 12 tracking failures (primarily due to incorrect associations), it was able to recover from each one within seconds. Overall, tracking failures only occurred 0.65% of the time for the proposed system, compared with 7.55% of the time for the map-matching system.

This simulation not only illustrates that improved accuracy can be obtained by using fixed sensors rather than on-board sensors, but furthermore demonstrates the proposed system’s intrinsic ability to recover quickly from tracking errors.

6 Field Performance Evaluation

Stability in tracking during crowded situations is another key requirement of this system. The previous evaluation demonstrated the basic accuracy of the system in a simulated environment, but in real field deployments, there may be multiple robots interacting with many people. The presence of people and other robots in the environment may contribute to increased occlusions, and the proximity of people to the robots and increased number of trajectories may contribute to ambiguities which could reduce the tracking performance of

our proposed system. To verify the performance of our system in such challenging conditions, we performed an evaluation of our system in the field, measuring its ability to track a team of four robots moving around and interacting with people in a crowded shopping arcade.

6.1 Environment

These evaluations were conducted 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.

This environment used the wired version of our tracking system. Six SICK laser range finders were installed in the space for human and robot tracking (see Fig. 11).

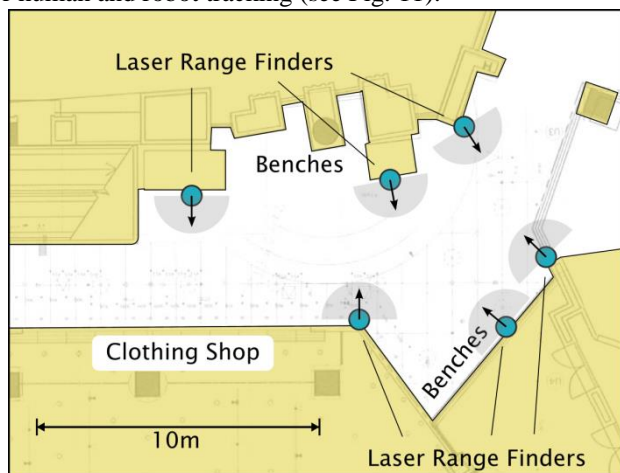


Fig. 11 Overview of the Universal CityWalk field environment. Six laser range finders monitored the central area, where robots approached customers and offered directions and recommendations.

6.2 Robot

For the evaluations presented here, Robovie II humanoid communication robots (Kanda et al. 2004b) were used. These robots consist of a humanoid upper body mounted on an Adept MobileRobots Pioneer 3 differential-drive platform for locomotion. For map generation and map-matching, the robot was also equipped with a Hokuyo UTM-30LX laser range finder.

6.3 Procedure

We ran four robots simultaneously in the field trial environment on a moderately crowded day for 30 minutes (roughly the amount of time we could continuously run the robots before changing the batteries) and recorded the tracking data for that period.

The robots were programmed to stop and interact with people who approached them, offering route guidance or shop recommendations to customers, as shown in Fig. 12. For these conversational interactions, the robot's dialogue was

teleoperated using the system presented in (Glas et al. 2012b). The teleoperator did not interact with the robots' localization in any way.

6.4 Measurement

In this evaluation, high-accuracy measurements of ground truth were unavailable. Instead, the recorded laser and tracking data were inspected manually to identify tracking errors. By inspection of the raw laser scan data, it is possible to estimate a robot's position by eye to within a few centimeters, and we recorded the distance between this position and the position estimated by the localization system.

We then applied the same criterion as we did in Sec. V, in which we defined "tracking errors" as data points where the robot's estimated position was at least 1 m away from its true position.



Fig. 12 Photo of four robots interacting with customers in the field environment.

6.5 Results

During the trial, the area was fairly crowded, with an average of 10 people tracked in the space at any time, along with the four robots. There were 4 instances of tracking errors, all of which were automatically corrected. In total, there were 7.94 seconds of tracking errors across the four robots during the 30-minute trial, a result of 99.89% accuracy. On average, each error was corrected within 2 seconds, and the longest of the four errors lasted for only 2.61 seconds.

Three of the errors occurred when the tracking system itself mistakenly switched the trajectories of the robot and a person standing very close by. In each case, the localization system successfully detected that the association was no longer valid, and it automatically reassociated the robot to the correct trajectory. The fourth error occurred in an area where tracking was briefly lost due to poor sensor coverage, and the robot was correctly reassociated after the tracking system redetected it.

In conclusion, the localization system worked quite stably. Even though there were many occlusions and the area was quite crowded, there were very few instances of tracking errors, and the localization system quickly compensated for errors made by the underlying tracking system.

7 Stability toward change in the environment

An important advantage of our technique over traditional map-based localization is its stability toward change in the environment. Environments such as shopping malls provide a particular challenge to localization because features of the environment change quite frequently, as product displays and vendor stands are often rearranged. For robots to provide service in these environments, robustness to such changes is a necessity.

We compared the proposed method with map matching in a real shopping mall environment for two reasons: first to confirm whether the differences observed in our simulation would also be seen in the real world; and second, to observe the robustness of map-matching to the changing features typical of a commercial environment like a shopping mall.

Ideally, when environmental features change, a new map should be created to provide the best localization accuracy; however, in a shopping mall, features can change significantly even within the span of a single day, whereas creating a new map can take a considerable amount of time and would be logistically impractical. In our experiment, we compared the performance of our system against two map-matching conditions: one using a current map, and one using a map built on a different day.

7.1 Experiment design

In our experiment, we manually drove the robot along an arbitrary path passing over twelve measured ground-truth points marked on the floor, stopping the robot at each point and verifying its location by eye. This process was conducted on two different days, when the product displays were in different arrangements, as shown in Fig. 13.

We then compared estimates of the robot's position using three estimation techniques. In the first condition, "LS," we used the outputs of the localization system proposed in this paper, which was run online while the robot drove around the environment. In the next condition, designated "MM1," we estimated the robot's location by offline map-matching with recorded odometry and laser data, using an environment map generated using data taken on the day of the trial. Finally, to illustrate the degree to which changing environments affect the accuracy of map-based localization, we repeated the offline map-matching using a map generated using data from the other day of the experiment, a condition we designated "MM2".

Our expectation was that whereas the map-matching localization approach using a recent map should provide highly accurate results, its accuracy would decrease as the configuration of objects in the environment diverged from that in the initial map. Thus, we expected that the MM2 condition would provide much lower accuracy than MM1.

Our localization system, on the other hand, is not map-based and thus is not sensitive to changes in the environment, so we expected the LS condition would provide slightly better performance than MM1, as our simulations suggested, and substantially better performance than MM2.

7.2 Environment

This evaluation was performed in Apita Town Keihanna, a shopping mall in Kyoto, Japan, where we have conducted several field studies in which mobile robots interact with shoppers (Iwamura et al. 2011),(Saiki et al. 2011)

The area studied in this evaluation was a space approximately 15 m wide and 30 m long in the central area of the mall, where a robot might be deployed to assist shoppers. Seasonal and event-oriented products are on display in this area, and their arrangement changes over time based on availability or events. Fig. 2 shows photos of this area taken on different days to illustrate this variation.

Some features in this space were fixed, such as columns, walls, service desks, and an escalator; whereas other features such as product displays changed on a daily, weekly, or seasonal basis.

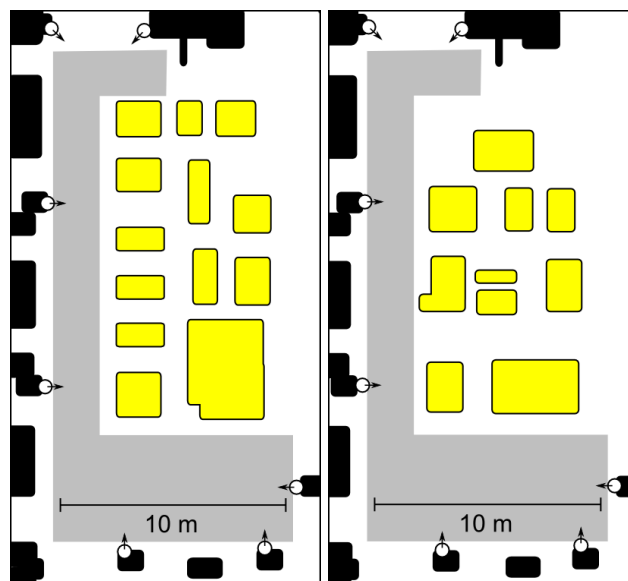


Fig. 13 Grid-maps of environment on two different days in a shopping mall. The features shown in black are permanent fixtures, whereas yellow features in the center are temporary product displays which are moved or replaced periodically. The gray area shows the region where the robot was driven, and white circles mark the environmental sensor positions.

In this environment, we set up the wireless variant of our tracking system using 7 Hokuyo UTM-30LX sensors (nominal range 30 m). These sensors were placed around the perimeter of the space.

We conducted our experiment during normal business hours at the shopping mall. The area was open to customers as usual, and people with shopping carts and baby strollers walked through the area during our testing, doing their everyday shopping. However, our experiments were conducted on weekday afternoons, during which time pedestrian traffic was very low. The robot did not interact with any customers in this experiment, and there were very few people in the environment and thus few occlusions which could interfere with either of the two localization techniques.

7.3 Evaluation of the proposed technique

For this evaluation, both the robot and the environmental sensors were connected by wireless LAN to a server running our localization software, which is our standard field-trial configuration. Range data from the sensors and odometry data from the robot were sent to the server every 30 ms, and the server sent localization corrections back to the robot at the same frequency. These corrected values were logged for later comparison with ground truth.

As a reference for measuring the accuracy of this technique, we marked 12 points on the floor and measured their locations to provide ground truth. We then drove the robot manually around the area, stopping it over these ground truth points. After visually confirming that the robot was aligned over a ground truth point, we logged both the robot's true position and its internal estimate based on our localization technique.

7.4 Map-based localization technique

This evaluation used the same map-matching technique used in the simulation experiment. On each day, we first generated a map by driving the robot slowly through the environment one time, turning it as little as possible. In offline processing, laser scan points representing moving objects such as people, shopping carts, and baby strollers were removed from the scan data. A grid map was then generated using scan data and odometry data from the robot using ICP-based SLAM.

To generate localization accuracy estimates for MM1 and MM2, we replayed laser scan data and odometry data in an offline simulator, again removing dynamic obstacles, and we used a particle filter to estimate the robot's position using a ray-tracing approach to estimate particle likelihood. Further details about the technique are presented in the Appendix.

7.5 Calibration

To provide a fair comparison of tracking accuracy, it was necessary to align both the generated scan maps and the human tracking system precisely to the ground-truth coordinate system. In both cases, this alignment was performed manually. For the scan maps, the robot was driven to and stopped at several ground truth points during the map-building run, and these points were used to align the coordinate systems in offline processing.

For the sensor network, registration with the ground-truth coordinate system was performed online before the experiment using detected positions of people standing at each of the ground truth points.

7.6 Data acquisition

On each day, we manually drove the robot for 30 minutes along an arbitrary path within this space. Throughout each trial, we recorded raw odometry, laser range data, and the coordinates provided by our localization system. At each reference point, the robot was stopped, and its position was

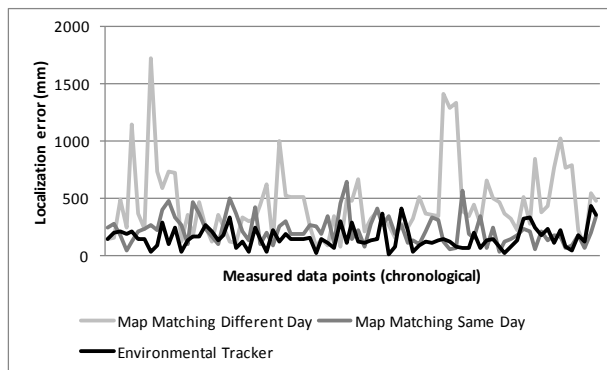


Fig. 14 Raw data of localization over time, comparing our proposed method with both sets of map-matching results.

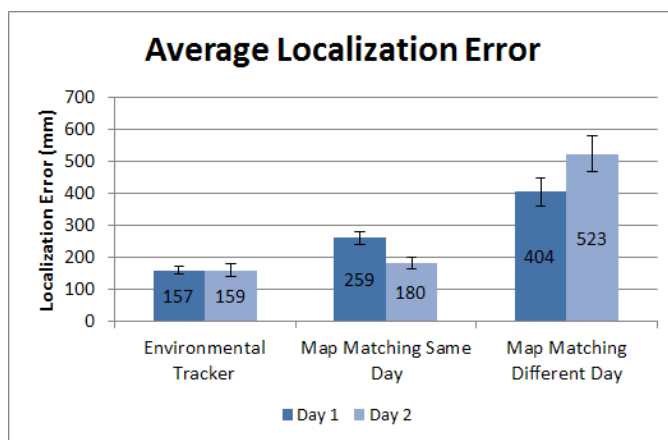


Fig. 15 Average localization error results from our field experiment.

verified by eye. If the position was within ± 2 cm of the reference mark, its coordinates and timestamp were logged. During each trial the robot drove over approximately 40 reference points.

This procedure was performed on two days, and then the offline map-matching estimates MM1 and MM2 were computed for both days of data.

7.7 Results

Between the two trials, we recorded a total of 81 instances of the robot driving over ground-truth reference points. The results of our error computations for the three measurement conditions are shown in Fig. 14 and summarized in Fig. 15, and the scatter plots in Fig. 16 show the absolute offset of each localization estimate with respect to ground truth.

The accuracy of our system (LS) was slightly better than that of the map-matching system with the same-day map (MM1), confirming the same trend we saw in the simulation data, although the magnitude of the difference was smaller.

The results of this experiment confirm not only that our proposed approach outperforms map-matching when an ideal same-day map is available, but also that the benefit of using our proposed technique rather than a map-based approach becomes greater as features in the environment diverge from the recorded map.

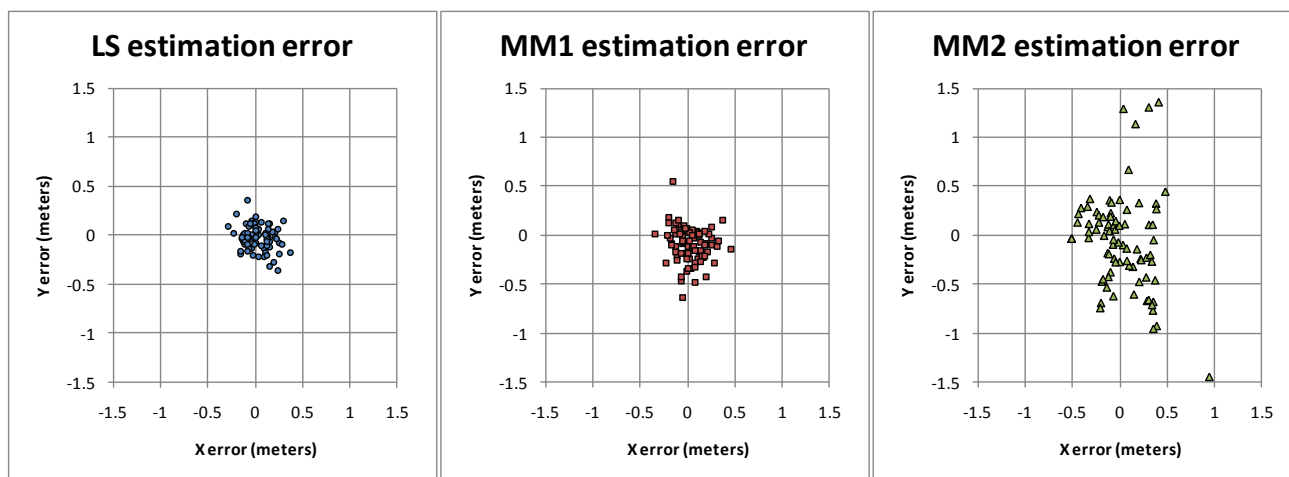


Fig. 16 Scatter plots of estimation error. Points shown represent absolute offset of each localization estimate from its corresponding ground truth reference point.

7.8 Discussion of results

7.8.1 Comparing field and simulation results

While the proposed method was more accurate than map matching in this field experiment, the difference was not as large as would be expected from the simulation results, i.e. comparing with the results in Fig. 10 (right). We believe that there are several possible explanations for this.

First, the positions of the sensors are not known perfectly. Our calibration system (Glas et al. 2010), has been demonstrated in the laboratory to estimate relative sensor positions within 40 mm and 0.5 degree error, but this performance varies according to the geometry of the sensor placement.

Second, the sensor mounting poles have become damaged over time and are not always perfectly level, and these pitch or roll offsets were not modeled in our tracking system. However, even considering these factors, the accuracy of our proposed system was still better than the map-matching case.

7.8.2 Frequency of feature changes in the environment

The localization error of the map-matching system was more than twice as high when using the different-day map (MM2), illustrating the sensitivity of a typical map-matching approach to changes in the environment.

Regarding the significance of this point, we would like to note that although we investigated the change of environment that occurs between two different days, it is also common for map features in a shopping mall to change even within the span of a single day. Examples include setting out and organizing new inventory in shop entrances, reorganizing products after customers have been browsing through or buying them, stacking full and empty crates as new inventory is unloaded, the highly variable number of shopping carts and shopping baskets available near the entrance to a supermarket, and time-dependent businesses such as lunchbox stands or placing signs in front of restaurants around meal times.

8 Discussion

8.1 Applicability

Our assumption of the existence of a sensor framework might seem unreasonable and limiting – after all, one cannot expect the entire world, or even an entire shopping mall, to be fully covered with embedded sensors.

We believe this is not a problem, as our proposed tracking and localization system is most useful in crowded spaces, where there will be many occlusions and distant people will not be visible to the robot, and in dynamic environments, like an entrance or market area where the vendor stalls and product displays are expected to change frequently. These are the areas where we believe it will be most useful to install external sensors. These also correspond to the areas where we believe social robots will most likely be deployed.

For other areas which are less crowded, or where the robot is not expected to interact heavily with people, we expect that such a sensor network might not be available. It may still be necessary for a robot to pass through these areas, but in these cases localization using on-board sensing should be sufficient.

8.2 Combination of approaches

Our proposed technique is designed such that if a robot is not associated to a tracked entity, no corrections are sent to the robot. This design makes it possible for the robot to rely on an on-board (e.g., map-based) localization system at times when it is not being tracked by the environmental sensors. Such a hybrid approach could take advantage of the strengths of both systems and be useful in scenarios where only part of an environment is covered by a tracking system.

In our field experiments we have used such a hybrid approach, relying on map-based localization when the robot is outside the areas covered by external sensors, and switching to the proposed system when the robots enter the covered areas.

8.3 Tracking systems in dynamic environments

We have shown that our technique outperforms map-matching when the environment changes, but this raises the question of how robust the tracking system is in dynamic environments.

Although our system does not use a map for localization, each sensor does have a model of the shape of the background environment. Unlike a robot, however, the sensors do not need to move in order to update this scan model – it can be automatically generated from a sequence of several scan frames. Typically we manually generate this in the morning after deploying the sensors, but we have also developed techniques for automatically detecting when a sensor has been moved or the environment changed, and for automatically calibrating the sensor positions (Glas et al. 2010; Glas et al. 2012a).

When pedestrian traffic is very heavy, occlusions can make reliable tracking difficult if only waist-level sensors are used. The use of ceiling-mounted RGB-D sensors rather than waist-height LRF's incurs a higher setup cost, but is more robust to occlusions (Brščić et al. 2013). As our proposed technique is independent of the specific tracking system used, it can work with ceiling-mounted systems as well.

8.4 Additional advantages

8.4.1 Non-invasive setup

In a real commercial space, it is important to avoid disrupting the shopping environment whenever possible. However, tasks such as map-building cannot be performed outside of business hours, because the features of the environment change significantly as soon as the shops open their shutters and put out their merchandise. Thus, accurate maps can only be built while the shops are open, a procedure which if repeated each day could be disruptive to the social environment. Our proposed sensor system is not affected by such changes in features, so it could be set up even before shops open for business.

8.4.2 Coordinate system registration

In any application where mobile robots must interact with people, it is necessary to guarantee that the robots are localized in the same coordinate system as the people being tracked. One convenient property of our system is that this registration occurs implicitly, as the same system is tracking all entities.

While not a theoretically difficult problem, it is a practical consideration in the deployment of systems in the field, and the use of the sensor system for tracking all people and robots greatly simplifies this task.

8.4.3 Ease of integration

An additional merit of our system is that it can easily accommodate new robots, as they do not need special hardware, *e.g.* to receive signals from beacons. In fact, robots

do not need any sensors at all for localization. Robots can take advantage of our localization system simply by adding a socket client that sends odometry data to our server and receives localization corrections. This functionality has come in quite useful when conducting joint field demonstrations with other research organizations where different robot types, such as robots with omnidirectional locomotion, are used.

8.5 Limitations

8.5.1 Robots outside the tracking area

The proposed technique assumes that every robot connected to the system is physically within the tracking area. The possibility that this assumption is violated (*e.g.*, a robot goes offline or drives outside of the tracking area) is not explicitly modeled in the proposed association algorithm. In our implementation, a robot must signal the system when it has entered or left the tracking area. This can be done autonomously if the robot has a rough estimate of its position.

8.5.2 Ambiguous situations

Regarding robustness when multiple robots are being tracked, we have observed the system to perform best when the robots are all moving. When multiple robots are stopped for extended periods of time, the system cannot distinguish between them based on their recent motion history. This ambiguity usually occurs at system startup, and associations are corrected as soon as the robots begin moving.

9 Conclusions

We have presented a novel system for localizing social robots using a network of laser range finders embedded in the environment. While the practice of placing sensors in the environment is not common for many robot applications, such a system can provide many benefits for social robot applications by providing information about pedestrians in the environment. In environments where such an infrastructure is available, our proposed technique provides several advantages over traditional map-based localization.

Simulation results show our approach to provide better tracking accuracy and faster, more reliable recovery from errors than a common map-matching localization technique using an on-board sensor. We have also demonstrated in the field that the system can localize several robots reliably in real time, even in high-occlusion situations when large numbers of people are present in the environment.

Finally, we have shown that our approach outperforms map-matching with on-board sensors in the field as well, and that the improvement provided by our system is particularly large when the map-matching system is using an outdated map in an environment where reference features change frequently.

While our proposed approach is not a replacement for techniques such as SLAM in the field of robotics in general, we believe it to be a valuable technique for social robots in

particular, and that it will enable the deployment of many kinds of social robot applications in active and dynamic everyday environments.

Appendix

Most mapping and localization approaches for mobile robots rely on the assumption that environments are static and do not change over time. Many dynamic obstacles such as pedestrians were present in our shopping mall environment, necessitating a method for removing moving objects from the robot's scan data. The next subsection explains this approach.

Obstacle removal

First we processed each scan of laser data, grouping points into clusters. When two consecutive points are within 0.1 m of distance they are grouped as a cluster. A covariance matrix is built from the point distribution of each cluster and the eigenvalues of the matrix are computed to analyze the shape of the cluster. If the length of a cluster is smaller than 0.5 m and the cluster distribution does not represent a straight line then it is determined that the cluster is a potential moving obstacle (one or two human legs together, a scan of a typical baby cart, or one of the simulated humans described in Section 5) and it is removed from the scan data. With this method small moving obstacles can be erased from the scan data for map building and localization. The drawback is that some small clusters which are part of the environment are removed as well; however, the noisy moving measurements are erased from the scene improving the resulting map and the localization performance.

Map building

As many variations of SLAM exist in the field of robot navigation, we aimed to use commonly-available tools and techniques for our comparison.

Map generation was performed by driving the robot slowly through the environment, avoiding frequent turns whenever possible. In our algorithm, we first erased from the laser scans small features, such as human legs, shopping carts, and baby strollers from the raw data, based on the cluster analysis criteria explained in the previous subsection. The laser scan and odometry data from the robot was recorded for this map-building run, and a raster map was built offline by an ICP-based SLAM to correct the trajectory of the robot and align the scans (Borrmann et al. 2008) using 3DTK SLAM software using 2D data (3DTK). With the resulting data we built a grid map (Moravec and Elfes 1985),(Elfes 1989) with the same coordinate system as the human tracking system.

Localization technique

To generate map-matching localization estimates for the experiment trials, we replayed laser scan data and odometry data in an offline simulator. Based on this data, our localization

algorithm used a particle filter to estimate the robot's position. The particle filter estimates the robot position with a weighted set of particles. The position of each particle is estimated using wheel encoder data. Particle likelihoods are updated using the previously constructed grid map with a ray tracing approach (Fox 2003),(Fox et al. 1999). In our implementation, we used 200 particles for the robot position estimation.

Acknowledgments Thanks to the staff of Universal CityWalk Osaka and Apita Town Keihanna for their cooperation in this research, and thanks to Dr. Satoshi Koizumi for his assistance in organizing the field trials.

References

- 3DTK 3DTK - The 3D Toolkit. <http://slam6d.sourceforge.net/>. Accessed July 22, 2013.
- Amarasinghe, D., Mann, G. K. I., & Gosine, R. G. Integrated laser-camera sensor for the detection and localization of landmarks for robotic applications. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on, 19-23 May 2008 2008* (pp. 4012-4017). amarasinghe08. doi:10.1109/robot.2008.4543827.
- Bennewitz, M., Burgard, W., Cielniak, G., & Thrun, S. (2005). Learning motion patterns of people for compliant robot motion. *The International Journal of Robotics Research*, 24(1), 31-48.
- Bevilacqua, A., Di Stefano, L., & Azzari, P. People Tracking Using a Time-of-Flight Depth Sensor. In *Video and Signal Based Surveillance, 2006. AVSS '06. IEEE International Conference on, Nov. 2006 2006* (pp. 89-89). doi:10.1109/avss.2006.92.
- Billard, A., Ijspeert, A. J., & Martinoli, A. (1999). A Multi-robot System for Adaptive Exploration of a Fast-changing Environment: Probabilistic Modeling and Experimental Study. *Connection Science*, 11(3-4), 359-379. doi:10.1080/095400999116304.
- Borrmann, D., Elseberg, J., Lingemann, K., Nüchter, A., & Hertzberg, J. (2008). Globally consistent 3D mapping with scan matching. *Robot. Auton. Syst.*, 56(2), 130-142. doi:10.1016/j.robot.2007.07.002.
- Bršćić, D., Kanda, T., Ikeda, T., & Miyashita, T. (2013). Person Tracking in Large Public Spaces Using 3-D Range Sensors. *Human-Machine Systems, IEEE Transactions on*, 43(6), 522-534. doi:10.1109/thms.2013.2283945.
- Burgard, W., Cremers, A. B., Fox, D., Hahnel, D., Lakemeyer, G., Schulz, D., et al. The interactive museum tour-guide robot. In *Proceedings of the fifteenth national/tenth conference on Artificial Intelligence/Innovative applications of artificial intelligence, Madison, Wisconsin, United States, 1998* (pp. 11-18). burgard98: American Association for Artificial Intelligence
- Burgard, W., Moors, M., Stachniss, C., & Schneider, F. E. (2005). Coordinated multi-robot exploration. *Robotics, IEEE Transactions on*, 21(3), 376-386. doi:10.1109/tro.2004.839232.
- Cui, J., Zha, H., Zhao, H., & Shibasaki, R. (2007). Laser-based detection and tracking of multiple people in crowds. *Comput. Vis. Image Underst.*, 106(2-3), 300-312. doi:10.1016/j.cviu.2006.07.015.
- Dissanayake, M. W. M. G., Newman, P., Clark, S., Durrant-Whyte, H. F., & Csorba, M. (2001). A solution to the simultaneous localization and map building (SLAM) problem. *Robotics and Automation, IEEE Transactions on*, 17(3), 229-241. doi:10.1109/70.938381.
- Elfes, A. (1989). Using occupancy grids for mobile robot perception and navigation. *Computer*, 22(6), 46-57. doi:10.1109/2.30720.
- Fod, A., Howard, A., & Mataric, M. A. J. A laser-based people tracker. In *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on, 2002 2002* (Vol. 3, pp. 3024-3029). fod02. doi:10.1109/robot.2002.1013691.
- Fox, D. (2003). Adapting the sample size in particle filters through KLD-sampling. *Intl. J. Robotics Res.*, 22, 985-1004. doi:10.1177/0278364903022012001.
- Fox, D., Burgard, W., & Thrun, S. (1999). Markov localization for mobile robots in dynamic environments. *Journal of Artificial Intelligence Research*, 2, 391-327.
- Gerkey, B., Vaughan, R. T., & Howard, A. (2003). The Player/Stage Project: Tools for Multi-Robot and Distributed Sensor Systems. In *11th*

- International Conference on Advanced Robotics (ICAR 2003), Coimbra, Portugal, June 2003* (pp. 317-323). gerkey03
- Glas, D. F., Ferreri, F., Miyashita, T., Ishiguro, H., & Hagita, N. (2012a). Automatic calibration of laser range finder positions for pedestrian tracking based on social group detections. *Advanced Robotics*.
- Glas, D. F., Kanda, T., Ishiguro, H., & Hagita, N. 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, 10-15 Oct. 2009 2009a* (pp. 846-853). glas09iros: IEEE Press. doi:10.1109/iros.2009.5354198.
- Glas, D. F., Kanda, T., Ishiguro, H., & Hagita, N. (2012b). Teleoperation of Multiple Social Robots. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 42(3), 530-544, doi:10.1109/tsmca.2011.2164243.
- Glas, D. F., Miyashita, T., Ishiguro, H., & Hagita, N. (2009b). Laser-Based Tracking of Human Position and Orientation Using Parametric Shape Modeling. *Advanced Robotics*, 23(4), 405-428, doi:10.1163/156855309x408754.
- Glas, D. F., Miyashita, T., Ishiguro, H., & Hagita, N. 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, 18-22 Oct. 2010 2010* (pp. 2938-2945). glas10iros. doi:10.1109/iros.2010.5652272.
- Gross, H.-M., Boehme, H., Schroeter, C., Müller, S., Koenig, A., Einhorn, E., et al. TOOMAS: interactive shopping guide robots in everyday use-final implementation and experiences from long-term field trials. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on, 2009* (pp. 2005-2012): IEEE
- Iwamura, Y., Shiomi, M., Kanda, T., Ishiguro, H., & Hagita, N. Do elderly people prefer a conversational humanoid as a shopping assistant partner in supermarkets? In *Proceedings of the 6th international conference on Human-robot interaction, Lausanne, Switzerland, 2011* (pp. 449-456). iwamura11: ACM. doi:10.1145/1957656.1957816.
- Kanda, T., Glas, D. F., Shiomi, M., & Hagita, N. (2009). Abstracting People's Trajectories for Social Robots to Proactively Approach Customers. *Robotics, IEEE Transactions on*, 25(6), 1382-1396, doi:10.1109/tro.2009.2032969.
- Kanda, T., Hirano, T., Eaton, D., & Ishiguro, H. (2004a). Interactive robots as social partners and peer tutors for children: A field trial. *Human-Computer Interaction*, 19(1), 61-84, doi:10.1207/s15327051hci1901&2_4.
- Kanda, T., Ishiguro, H., Imai, M., & Ono, T. (2004b). Development and evaluation of interactive humanoid robots. *Proceedings of the IEEE*, 92(11), 1839-1850, doi:10.1109/jproc.2004.835359.
- Kanda, T., Shiomi, M., Miyashita, Z., Ishiguro, H., & Hagita, N. (2010). A communication robot in a shopping mall. *Trans. Rob.*, 26(5), 897-913, doi:10.1109/tro.2010.2062550.
- Köse, H., & Akin, H. L. (2007). The Reverse Monte Carlo localization algorithm. *Robot. Auton. Syst.*, 55(6), 480-489, doi:10.1016/j.robot.2006.12.007.
- Lee, J.-S., & Chung, W. K. (2010). Robust mobile robot localization in highly non-static environments. *Auton. Robots*, 29(1), 1-16, doi:10.1007/s10514-010-9184-1.
- Montemerlo, M., Thrun, S., & Whittaker, W. Conditional particle filters for simultaneous mobile robot localization and people-tracking. In *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on, 2002 2002* (Vol. 1, pp. 695-701 vol.691). montemerlo02. doi:10.1109/robot.2002.1013439.
- Moravec, H., & Elfes, A. High resolution maps from wide angle sonar. In *Robotics and Automation. Proceedings. 1985 IEEE International Conference on, Mar 1985 1985* (Vol. 2, pp. 116-121). moravec85. doi:10.1109/robot.1985.1087316.
- Mutlu, B., & Forlizzi, J. Robots in organizations: the role of workflow, social, and environmental factors in human-robot interaction. In *Proceedings of the 3rd ACM/IEEE international conference on Human robot interaction, Amsterdam, The Netherlands, 2008* (pp. 287-294). mutlu08: ACM. doi:10.1145/1349822.1349860.
- Ouellette, R., & Hirasawa, K. A comparison of SLAM implementations for indoor mobile robots. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on, Oct. 29 2007-Nov. 2 2007 2007* (pp. 1479-1484). ouellette07. doi:10.1109/iros.2007.4399575.
- Pacchierotti, E., Christensen, H. I., & Jensfelt, P. Design of an Office-Guide Robot for Social Interaction Studies. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on, 9-15 Oct. 2006 2006* (pp. 4965-4970). pacchierotti06. doi:10.1109/iros.2006.282519.
- Park, S., Saegusa, R., & Hashimoto, S. Autonomous navigation of a mobile robot based on passive RFID. In *Robot and Human interactive Communication, 2007. RO-MAN 2007. The 16th IEEE International Symposium on, 26-29 Aug. 2007 2007* (pp. 218-223). park07. doi:10.1109/roman.2007.4415083.
- Pizarro, D., Marron, M., Peon, D., Mazo, M., Garcia, J. C., Sotelo, M. A., et al. Robot and obstacles localization and tracking with an external camera ring. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on, 19-23 May 2008 2008* (pp. 516-521). pizarro08. doi:10.1109/robot.2008.4543259.
- Saffiotti, A., Broxvall, M., Gritti, M., LeBlanc, K., Lundh, R., Rashid, J., et al. The PEIS-Ecology project: Vision and results. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on, 22-26 Sept. 2008 2008* (pp. 2329-2335). saffiotti08. doi:10.1109/iros.2008.4650962.
- Saiki, L. Y. M., Satake, S., Kanda, T., & Hagita, N. Modeling environments from a route perspective. In *Proceedings of the 6th international conference on Human-robot interaction, Lausanne, Switzerland, 2011* (pp. 441-448). luis11: ACM. doi:10.1145/1957656.1957815.
- Satake, S., Kanda, T., Glas, D. F., Imai, M., Ishiguro, H., & Hagita, N. (2010). How to Approach Humans? : Strategies for Social Robots to Initiate Interaction. *Journal of the Robotics Society of Japan*, 28(3), 327-337.
- Schulz, D., Burgard, W., Fox, D., & Cremers, A. B. (2003). People Tracking with Mobile Robots Using Sample-Based Joint Probabilistic Data Association Filters. *The International Journal of Robotics Research*, 22(2), 99-116, doi:10.1177/0278364903022002002.
- Schulz, D., Fox, D., & Hightower, J. People tracking with anonymous and ID-sensors using Rao-Blackwellised particle filters. In *International joint conference on artificial intelligence, 2003* (Vol. 18, pp. 921-928). schulz03rbpf: LAWRENCE ERLBAUM ASSOCIATES LTD
- Siegrwart, R., Arras, K. O., Bouabdallah, S., Burnier, D., Froidevaux, G., Greppin, X., et al. (2003). Robox at Expo.02: A large-scale installation of personal robots. *Robotics and Autonomous Systems*, 42(3), 203-222, doi:10.1016/s0921-8890(02)00376-7.
- Sisbot, E. A., Alami, R., Simeon, T., Dautenhahn, K., Walters, M., & Woods, S. Navigation in the presence of humans. In *Humanoid Robots, 2005 5th IEEE-RAS International Conference on, 5-5 Dec. 2005 2005* (pp. 181-188). sisbot05. doi:10.1109/hchr.2005.1573565.
- Takahashi, T. (2007). *2D Localization of Outdoor Mobile Robots Using 3D Laser Range Data*. Master's Thesis, Carnegie Mellon University,
- Thrun, S., Bücken, A., Burgard, W., Fox, D., Frölinghaus, T., Hennig, D., et al. (1998). Map learning and high-speed navigation in RHINO. In *Artificial intelligence and mobile robots* (pp. 21-52): MIT Press.
- Thrun, S., Fox, D., Burgard, W., & Dallaert, F. (2001). Robust Monte Carlo localization for mobile robots. *Artif. Intell.*, 128(1-2), 99-141, doi:10.1016/s0004-3702(01)00069-8.
- Urmson, C., Anhalt, J., Bagnell, D., Baker, C., Bittner, R., Clark, M., et al. (2008). Autonomous driving in urban environments: Boss and the urban challenge. *Journal of Field Robotics*, 25(8), 425-466.
- Vu, T.-D., Aycaud, O., & Appenrodt, N. Online Localization and Mapping with Moving Object Tracking in Dynamic Outdoor Environments. In *Intelligent Vehicles Symposium, 2007 IEEE, 13-15 June 2007 2007* (pp. 190-195). vu07. doi:10.1109/ivs.2007.4290113.
- Wang, C.-C., Thorpe, C., & Thrun, S. Online simultaneous localization and mapping with detection and tracking of moving objects: theory and results from a ground vehicle in crowded urban areas. In *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on, 14-19 Sept. 2003 2003* (Vol. 1, pp. 842-849 vol.841). wang03. doi:10.1109/robot.2003.1241698.
- Wolf, D. F., & Sukhatme, G. S. (2005). Mobile Robot Simultaneous Localization and Mapping in Dynamic Environments. *Auton. Robots*, 19(1), 53-65, doi:10.1007/s10514-005-0606-4.
- Zhao, H., Chiba, M., Shibasaki, R., Shao, X., Cui, J., & Zha, H. SLAM in a dynamic large outdoor environment using a laser scanner. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on, 19-23 May 2008 2008* (pp. 1455-1462). zhao08. doi:10.1109/robot.2008.4543407.