

# *Sancta Simplicitas* – On the efficiency and achievable results of SLAM using ICP-Based Incremental Registration

Dirk Holz and Sven Behnke

**Abstract**—This paper presents an efficient combination of algorithms for SLAM in dynamic environments. The overall approach is based on range image registration using the ICP algorithm. Different extensions to this algorithm are used to incrementally construct point models of the robot’s workspace. A simple heuristic allows for determining which points in a newly acquired range image are already contained in the point model and for adding only those points that provide new information. Furthermore, the means for dealing with environment dynamics are presented which allow for continuously conducting SLAM and updating the point model according to changes in a dynamic environment. The achievable results of the overall approach are compared to Rao-Blackwellized Particle Filters as a state-of-the-art solution to the SLAM problem and evaluated using a recently published benchmark by Burgard *et al.* (2009).

## I. INTRODUCTION

Autonomous service robots face the challenging task of operating in real-world indoor and domestic environments. Domestic environments tend to be cluttered, dynamic and populated by humans and domestic animals. In order to adequately react to sudden dynamic changes and avoid collisions, these robots need to be able to constantly acquire and process information about their environment in real-time. Furthermore, in order to act in a goal-directed manner, plan actions and navigate effectively, a robot needs an internal representation of its environment.

Especially in cluttered and dynamic environments it is uncomfortable, even unfeasible, to provide the robot with a manually constructed environment model. Simultaneous Localization and Mapping (SLAM) forms a major precondition for truly autonomous robots and provides the means for minimizing the amount of information that has to be manually given to the robot. A robot performing SLAM has, in principal, the ability to construct an internal environment representation or *map* on its own. These maps then cover all environmental structures that have been sensed at least once. An approach that is widely used in real-world applications is to manually steer the robot around, collect all acquired sensory information and process the gathered information offline in order to construct an environment model that can be used in subsequent tasks. However, due to the fact that an autonomous service robot has to operate in a dynamic environment, it must also constantly acquire information about the environment during later navigation tasks, and use this information to update the map. Furthermore, there are

several applications where the robot needs to construct an initial environment model on the fly during operation, e.g., when operating in a previously unknown environment. Here SLAM has to be addressed as a continuous lifelong problem. The used algorithms need to be efficient so that they can be applied on the robot during operation and process acquired sensory information in real-time.

Over the last two decades, different algorithms for addressing the SLAM problem have been proposed. In recent years, there is a trend to probabilistic SLAM algorithms using, for example, Extended Kalman Filters (EKF) [1], Unscented Kalman Filters (UKF) [2], Sparse Extended Information Filters (SEIF) [3] or Rao-Blackwellized Particle Filters (RBPF) [4]. These explicitly handle uncertainties about the conducted estimates and the processed sensory information by estimating a probability distribution over the possible solutions. While they achieve robust and accurate results, the involved computational effort often prevents their application to large problem instances and hinders real-time applicability.

Graph-based SLAM algorithms [5], [6] formulate the problem in terms of a pose graph where edges pose constraints on the relation between vehicles poses (the nodes in the graph). Highly efficient optimization algorithms are used to find configurations of nodes/poses that fulfill and are maximally probable given the constraints in the graph. Since determining the constraints in the graph involves a data association problem, these approaches are often used as generic back-ends in a SLAM approach with matching (and feature extraction) algorithms in a front-end that determine the corresponding associations.

Especially in the context of 3D-SLAM for constructing three-dimensional environment maps and localizing the robot with six degrees of freedom, scan matching algorithms are used to register multiple range images and to determine the relations between the poses where the range images have been taken [7], [8]. Under certain assumptions, e.g., that loops in the robot’s trajectory are not too large, the pose relations determined by registration algorithms can be accurate enough for constructing globally consistent environment models. One of the most prominent algorithms for range image registration, the Iterative Closest Point (ICP) algorithm will also be used in the proposed approach.

There are different ways for registering multiple range scans. The most prominent one is pairwise registration where a newly acquired range scan is matched against the last range scan in order to determine the change in position and orientation between the poses where the scans have been taken. This procedure is commonly found in preprocessing

steps of SLAM algorithms to correct odometric pose shift estimates. Since an individual registration considers only two range scans, this procedure is computationally very efficient. However, when used for estimating the robot’s trajectory and mapping the environment, registration errors accumulate and can lead to inconsistencies in the resulting map.

Another widely used post-processing technique for distributing and minimizing residual errors is to register multiple range scans against each other until the determined pose shifts do no longer change [9]. The registration of a newly acquired range scan is followed by registering all neighboring scans. Taking this additional information into account allows for correcting inconsistencies but also increases the complexity of the registration procedure.

Here, we apply an incremental registration procedure against a so-called *meta-scan* [10]. Just like in the pairwise procedure, registration errors accumulate. However, by aggregating the so far gathered information in the meta-scan (the actually built point map), the robot is localized in the map and accumulated registration errors are corrected within the current pose shift estimate. Since the integration of a newly acquired range scan involves only a single registration against the meta-scan, it is rather computationally efficient (compared to [9]) and scales logarithmically with the size of the meta-scan and the constructed point map, respectively (see Section II). If the accumulated error before a correction is not too large, incremental registration can construct globally consistent point maps without further post-processing (see Figure 1 (a+b)).

The remainder of this paper is organized as follows: the next section describes the used incremental registration procedure together with a detailed description of the ICP algorithm as well as the derivation of a closed-form solution for the two-dimensional case and the extensions that have been used to construct globally consistent 2D point maps. Section III compares the achieved results with those of a widely used Rao-Blackwellized particle filter implementation [4] using a recently published benchmark [11].

## II. INCREMENTAL REGISTRATION USING THE ICP

This section describes the proposed approach including the ICP algorithm and the used extensions to speed up correspondence search, neglect false correspondences and remove or filter out measurements belonging to dynamic objects, like for instance people passing by.

### A. The ICP Algorithm

Given two data sets, a model set  $M$  and a data set  $D$ , the ICP algorithm iteratively refines a transformation  $\mathbf{T} = (\mathbf{R}_{\Delta\theta}, \mathbf{t})$  by conducting (in principal) the following three steps:

- 1) Determine correspondences between  $D$  and  $M$ .
- 2) Determine the alignment error  $E$  between the corresponding instances.
- 3) Minimize the alignment error  $E$  or stop if some termination criterion is met.

$T$  thereby consists of a rotation  $\mathbf{R}_{\Delta\theta}$  by an angle  $\Delta\theta$  about the  $z$ -axis and a translation  $\mathbf{t} = (\mathbf{t}^x \ \mathbf{t}^y)^T$  along the  $x$ - and  $y$ -axes. Variations of and extensions to the ICP algorithm address the metric for measuring distances as well as the minimization of the alignment error  $E$ , rejecting and weighting the correspondences or improving the algorithm’s performance and robustness [12]. Besl and McKay [13] break down the registration of arbitrary geometric features into aligning the points defining these instances and formulate  $E$  in terms of a point-to-point metric. Another commonly found way of expressing the alignment error  $E$  is the point-to-plane metric [10]. Since the point-to-plane metric necessitates information about the local surface structure that is not directly available in unordered point sets, we only consider the point-to-point metric and define  $E$  as

$$E(\mathbf{R}, \mathbf{t}) = \sum_i^{|D|} \sum_j^{|M|} w_{ij} \|\mathbf{m}_j - \underbrace{(\mathbf{R}_n \mathbf{d}_i + \mathbf{t}_n)}_{\check{\mathbf{d}}_{i,n}}\|^2, \quad (1)$$

where  $w_{ij}$  is 1 if  $d_j$  and  $m_i$  correspond to the same physical point in the real environment (assuming no further weighting) and 0 if not. The correspondences are determined by a nearest neighbor search. That is, the only information that is needed when using the point-to-point metric is the closest point in  $M$  for every point in the data set  $D$  that is to be registered. Under the above assumptions and given a set of pairs of corresponding points  $\{(\check{\mathbf{d}}_k, \check{\mathbf{m}}_k)\}$ ,  $\mathbf{T}$  results from minimizing

$$\mathbf{T} = \arg \min_{(\mathbf{R}_{\Delta\theta}, \mathbf{t})} \sum_k \|\check{\mathbf{m}}_k - (\mathbf{R}_{\Delta\theta} \check{\mathbf{d}}_k + \mathbf{t})\|^2. \quad (2)$$

By considering variants  $\hat{M} = \{\hat{\mathbf{m}}_i \mid \hat{\mathbf{m}}_i = \check{\mathbf{m}}_i - \check{\mathbf{c}}_m\}$  and  $\hat{D} = \{\hat{\mathbf{d}}_i \mid \hat{\mathbf{d}}_i = \check{\mathbf{d}}_i - \check{\mathbf{c}}_d\}$  of the point sets  $\check{M}$  and  $\check{D}$ , both being shifted by their centroids  $\check{\mathbf{c}}_M$  and  $\check{\mathbf{c}}_D$ , determining the rotation  $\mathbf{R}_{\Delta\theta}$  can be decoupled from determining the translation  $\Delta\mathbf{t}$  [14]:

$$E(\mathbf{R}_{\Delta\theta}, \mathbf{t})^{(1)} = \sum_{i=1}^N \|\hat{\mathbf{m}}_i - \mathbf{R}_{\Delta\theta} \hat{\mathbf{d}}_i - \underbrace{(\mathbf{t} - \check{\mathbf{c}}_M + \mathbf{R}_{\Delta\theta} \check{\mathbf{c}}_D)}_{=\tilde{\mathbf{t}}}\|^2 \quad (3a)$$

$$= \sum_{i=1}^N \|\hat{\mathbf{m}}_i - \mathbf{R}_{\Delta\theta} \hat{\mathbf{d}}_i\|^2 \quad (3b)$$

$$- 2\tilde{\mathbf{t}} \cdot \sum_{i=1}^N (\hat{\mathbf{m}}_i - \mathbf{R}_{\Delta\theta} \hat{\mathbf{d}}_i) \quad (3b)$$

$$+ \sum_{i=1}^N \|\tilde{\mathbf{t}}\|^2. \quad (3c)$$

The second term (3b) is 0 since  $\hat{\mathbf{m}}_i$  and  $\hat{\mathbf{d}}_i$  are referred to the respective centroids. Since term (3c) cannot be negative and finds its minimum at  $\tilde{\mathbf{t}} = 0$ , the optimal translation  $\mathbf{t}$  is

$$\mathbf{t} = \check{\mathbf{c}}_m - \mathbf{R}_{\Delta\theta} \check{\mathbf{c}}_d. \quad (4)$$

Furthermore, with  $\tilde{\mathbf{t}} = 0$  Eq. (3) can be minimized by minimizing the first term (3a) which is not depending on

the translation  $\mathbf{t}$ :

$$E(\mathbf{R}_{\Delta\theta})^{(2)} = \underbrace{\sum_{i=1}^N \|\hat{\mathbf{m}}_i\|^2}_{\Sigma_{\hat{M}}} - 2 \underbrace{\sum_{i=1}^N (\hat{\mathbf{m}}_i \cdot \mathbf{R}_{\Delta\theta} \hat{\mathbf{d}}_i)}_{\Sigma_{\hat{M}\hat{D}}} + \underbrace{\sum_{i=1}^N \|\mathbf{R}_{\Delta\theta} \hat{\mathbf{d}}_i\|^2}_{\Sigma_{\hat{D}}}. \quad (5)$$

Since rotation is length preserving, i.e.  $\|\mathbf{R}_{\Delta\theta} \hat{\mathbf{d}}_i\|^2 \equiv \|\hat{\mathbf{d}}_i\|^2$ ,  $\Sigma_{\hat{M}}$  and  $\Sigma_{\hat{D}}$  cannot be negative and a maximization of  $\Sigma_{\hat{M}\hat{D}}$  minimizes  $E(\mathbf{R}_{\Delta\theta})^{(2)}$ . In the 2D case, with  $\Sigma_{\hat{M}\hat{D}}$  being

$$\Sigma_{\hat{M}\hat{D}}(\Delta\theta) = \cos\Delta\theta \sum_{i=1}^N (\hat{\mathbf{m}}_i^x \hat{\mathbf{d}}_i^x + \hat{\mathbf{m}}_i^y \hat{\mathbf{d}}_i^y) + \sin\Delta\theta \sum_{i=1}^N (\hat{\mathbf{m}}_i^y \hat{\mathbf{d}}_i^x - \hat{\mathbf{m}}_i^x \hat{\mathbf{d}}_i^y) \quad (6)$$

$\Delta\theta$  and  $\mathbf{t} = (\mathbf{t}^x \ \mathbf{t}^y)^T$  can be resolved to:

$$\Delta\theta = \arctan\left(\frac{\sum_{i=1}^N (\hat{\mathbf{m}}_i^y \hat{\mathbf{d}}_i^x - \hat{\mathbf{m}}_i^x \hat{\mathbf{d}}_i^y)}{\sum_{i=1}^N (\hat{\mathbf{m}}_i^x \hat{\mathbf{d}}_i^x + \hat{\mathbf{m}}_i^y \hat{\mathbf{d}}_i^y)}\right) \quad (7)$$

$$\mathbf{t} = \begin{pmatrix} \check{\mathbf{c}}_m^x \\ \check{\mathbf{c}}_m^y \end{pmatrix} \cdot \begin{pmatrix} \cos\Delta\theta & -\sin\Delta\theta \\ \sin\Delta\theta & \cos\Delta\theta \end{pmatrix} \begin{pmatrix} \check{\mathbf{c}}_d^x \\ \check{\mathbf{c}}_d^y \end{pmatrix}. \quad (8)$$

That is, the original point-to-point ICP algorithm (in the remainder referred to as *Vanilla ICP*) can be formulated as:

**Algorithm 1:** ICP algorithm with point-to-point metric.

---

**Input:**  $M = \{m_i\}, D = \{d_j\}, \mathbf{T}_0 = (\mathbf{R}_{\theta_0}, (\mathbf{t}^{x_0} \ \mathbf{t}^{y_0})^T)$   
**Output:**  $\mathbf{T} = (\mathbf{R}_{\Delta\theta}, \mathbf{t}), \{\mathbf{T}\}D$   
 $\mathbf{T} = \mathbf{T}_0, n = 0, \{\mathbf{T}\}D = D, (E_0(\mathbf{R}, \mathbf{t}) = \infty, \Delta E = \infty;$   
**while**  $(E_n(\mathbf{R}_{\Delta\theta}, \mathbf{t}) > \varepsilon_E) \wedge (\Delta E > \varepsilon_{\Delta E}) \wedge (n < n_{max})$  **do**  
     $n = n + 1;$   
     $\{\mathbf{T}\}D = \{\mathbf{R}_{\Delta\theta} \mathbf{d}_j + \mathbf{t} \mid \mathbf{d}_j \in D\};$   
     $E_n(\mathbf{R}_{\Delta\theta}, \mathbf{t}) = 0;$   
    **foreach**  $\{\mathbf{T}\}d_i \in \{\mathbf{T}\}D$  **do**  
         $\check{\mathbf{m}}_i = \arg \min_{\mathbf{m}_j \in M} \|\mathbf{m}_j - \{\mathbf{T}\}d_i\|;$   
         $E_n(\mathbf{R}_{\Delta\theta}, \mathbf{t}) = E_n(\mathbf{R}_{\Delta\theta}, \mathbf{t}) + \|\mathbf{m}_j - \{\mathbf{T}\}d_i\|;$   
    **end**  
     $\Delta E = E_{n-1}(\mathbf{R}_{\Delta\theta}, \mathbf{t}) - E_n(\mathbf{R}_{\Delta\theta}, \mathbf{t});$   
    **if**  $E(\mathbf{R}_{\Delta\theta}, \mathbf{t}) > \varepsilon_E$  **then**  
         $\mathbf{T}_{ICP} = \arg \min_{(\mathbf{R}_{\Delta\theta}, \mathbf{t})} \sum_k \|\check{\mathbf{m}}_k - (\mathbf{R}_{\Delta\theta} \check{\mathbf{d}}_k + \mathbf{t})\|^2;$   
         $\mathbf{T} = \mathbf{T}_{ICP} \oplus \mathbf{T}$   
    **end**  
**end**

---

where  $\{\mathbf{T}\}D$  refers to  $D$  transformed into the coordinate frame expressed by  $\mathbf{T}$ . The operator  $\oplus$  corresponds to applying the estimated transformation on and refining the transformation  $\mathbf{T}$ . The algorithm terminates when 1.) the alignment error  $E(\mathbf{R}_{\Delta\theta}, \mathbf{t})$  falls below the threshold  $\varepsilon_E$ , 2.) it has converged ( $\Delta E \leq \varepsilon_{\Delta E}$ ) or 3.) the maximum number of iteration steps  $n_{max}$  has been reached.

The computationally most expensive step in the ICP algorithm is the correspondence search. We need to find the closest point in  $M$  for every point in  $D$  which takes  $O(|D||M|)$  for a naïve implementation. As already suggested in [13] we use an algorithm [15] for approximate nearest neighbor search in a  $kd$ -tree of  $M$  that reduces the algorithm's complexity to  $O(|D|\log|M|)$ .

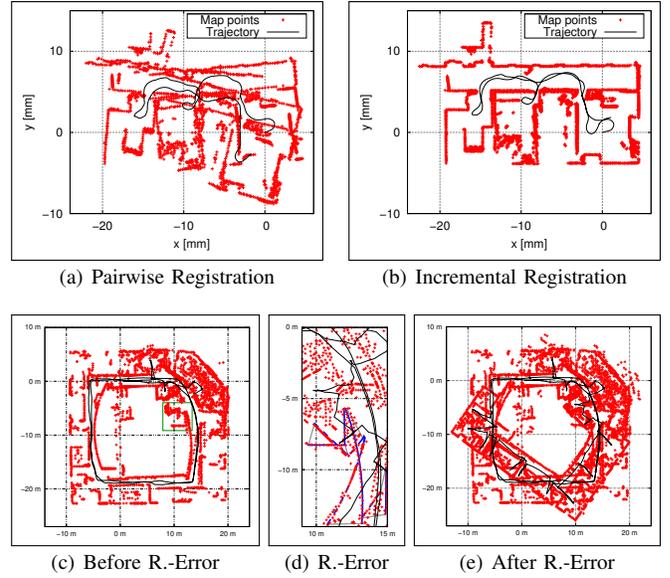


Fig. 1. TOP: Pairwise and incremental matching on a dataset by Minguez *et al.* [16]. With incremental registration all the information being acquired so far is taken into account when localizing the robot (smaller accumulated registration errors are corrected). BOTTOM: Larger registration errors (*R-Error* in c-e) can cause global inconsistencies, e.g., two locally consistent maps that are distorted against each other.

## B. Incremental Registration

In the incremental registration procedure, the first laser scan  $D_0$  is used as the initial environment model  $M_0$  and forms the coordinate frame for the overall map. All subsequent scans  $D_i, i > 0$  are matched against  $M_{i-1}$ . The resulting transformation  $\mathbf{T}_i$  is used to correct the position of all points contained in  $D_i$ , yielding the transformed point set  $\check{D}_i = \{\check{\mathbf{d}}_{i,j} \mid \check{\mathbf{d}}_{i,j} = \mathbf{R} \mathbf{d}_{i,j} + \mathbf{t}\}$ .

As an initial estimate  $\hat{\mathbf{T}}_i$  for  $\mathbf{T}_i$  in this incremental registration we use the transformation from the last registration, i.e.  $\hat{\mathbf{T}}_i = \mathbf{T}_{i-1}$ . To account for the robot's movement between poses  $\mathbf{x}_{i-1}$  and  $\mathbf{x}_i$  where the range scans  $D_{i-1}$  and  $D_i$  have been acquired,  $\hat{\mathbf{T}}_i$  is further updated using the odometric pose shift estimate  $\hat{\delta}_{i,i-1} = \hat{\mathbf{x}}_i \ominus \hat{\mathbf{x}}_{i-1}$ . This procedure is comparable to the motion updates in a particle filter with using  $\mathbf{x}_{i-1} \oplus \hat{\delta}_{i,i-1}$  instead of sampling from the motion model  $P(\mathbf{x}_{i-1} \oplus \hat{\delta}_{i,i-1} \mid \hat{\delta}_{i,i-1}, \mathbf{x}_{i-1})$ . It speeds up the convergence in the ICP algorithm and drastically reduces the probability of converging to a local minimum which could possibly result in an incorrect registration result. Furthermore, we only register a new range scan  $D_i$  if the robot traversed more than a certain distance (e.g. 30 cm) or turned more than a certain angle (e.g. 15°).

To account for possibly new information in  $D_i$ , the transformed points are then added to  $M_{i-1}$ . That is, after matching range image  $D_i$ , the model set  $M_{i-1}$  computed so far is updated in step  $i$  to:

$$M_i = M_{i-1} \cup \{\check{\mathbf{d}}_{i,j} \mid \check{\mathbf{d}}_{i,j} \in \check{D}_i\}. \quad (9)$$

A model constructed by the method described so far contains all points measured in the environment.

### C. Sparse Point Maps

The inherent problem of this procedure is its scalability with respect to the size of the environment and the number of range images taken. When registering and adding all acquired range measurements to the map, the model set  $M$  can grow quite large, e.g. several million points for 3D scans taken in a large outdoor environment [7]. However, when acquiring range images in parts of the environment which are already mapped, lots of points would be added to  $M$  without providing new information about the environment. This is exploited by the following extensions to the incremental registration procedure, which makes the point clouds sparse.

The key idea of *sparse point maps* is to avoid duplicate storage of points by conducting an additional correspondence search and neglecting points that correspond to the same point in the real physical environment as a point already stored in the map. Correspondence is thereby defined just like in the ICP algorithm. That is, a point  $\check{\mathbf{d}}_{i,j} \in \check{D}_i$  is not added to  $M_{i-1}$ , if the point-to-point distance to its closest point  $\mathbf{m}_{i-1,k} \in M_{i-1}$  is smaller than a minimum allowable distance  $d_{\min}$ :

$$M_i = M_{i-1} \cup \{\check{\mathbf{d}}_{i,j} \mid \nexists \mathbf{m}_{i-1,k} \in M_{i-1} : \|\check{\mathbf{d}}_{i,j} - \mathbf{m}_{i-1,k}\| < d_{\min}\}. \quad (10)$$

The threshold  $d_{\min}$  spans regions in the model in which the number of points is limited to 1, thereby providing an upper bound on the point density in a sparse point map  $M$ . Choosing a value of  $d_{\min}$  according to the accuracy of the range sensor will exactly neglect duplicate storage of one and the same point assuming correct alignment of range images. Choosing, however, a larger value allows to reduce the number of points stored in the map. Although some details of the environment might not get modeled, a map constructed in this manner still provides a coarse-grained model of the environment. In the actual implementation, the additional correspondence search is carried out on the same  $kd$ -tree built for the ICP algorithm.

### D. Additional Heuristics for Improving Registration Results

As can be seen in Figure 1, a single misregistration can cause a major map inconsistency. Most often these registration errors are caused by incorrect or poor odometric estimates, a larger number of invalid measurements in the range scans or a larger number of false correspondences ( $\check{\mathbf{d}}_{i,j}, \check{\mathbf{m}}_k$ ). Whereas the latter is being dealt with by rejecting certain correspondence pairs (see II-E), we apply the following simple heuristics for handling the first two problems.

- 1) If the odometric pose shift estimate  $\hat{\delta}_{i,i-1} = \hat{\mathbf{x}}_i \ominus \hat{\mathbf{x}}_{i-1}$  shows larger jumps, we stop using  $\hat{\delta}_{i,i-1}$  in the estimate  $\hat{\mathbf{T}}_i$  and switch into a *pose tracking mode*. In this mode we determine  $\delta_{i,i-1}$  by registering consecutive scans in a pairwise fashion (not only after travelling a certain distance). As soon as the odometric pose shifts are reasonable again, we switch back to using odometry in the initial estimate and only register a scan when the aforementioned thresholds are reached. By this means

larger odometry errors causing major misregistrations are normally ignored.

- 2) If the number of corresponding points in the incremental registration falls below some threshold (e.g. 30% of  $|D|$ ) or the determined pose shift  $\delta_{i,i-1} = \mathbf{T}_i - \mathbf{T}_{i-1}$  considerably deviates from the pose shift estimate  $\hat{\delta}_{i,i-1}$ , we do not use  $\mathbf{T}_i$  but continue with the estimate  $\hat{\mathbf{T}}_i$ . That is, we trust the odometric pose shift estimate. Conducted experiments have shown that the involved inaccuracies causing small deviations from the true trajectory are cleared in later successful registrations.

### E. Estimating Overlap and Rejecting Pairs

The original formulation of the ICP algorithm [13] assumes that there is a corresponding point in  $M$  for every point in  $D$ . When registering scans taken at different vehicle poses this is typically not the case and the two point sets overlap only partially. A common workaround is to reject correspondence pairs whose point-to-point distance exceeds some threshold  $d_{\max}$ . A larger value for  $d_{\max}$  accounts for poor initial estimates and improves initial convergence, but also causes false correspondences leading to less accurate registration results, compared to smaller values of  $d_{\max}$ . Instead of using a constant threshold, we let  $d_{\max}$  exponentially decay during the registration process. While initially permitting larger distances between corresponding points (e.g. 2m) guarantees fast convergence of  $E(R,t)$ , smaller distances (e.g. 10cm) in later iteration steps allow fine-tuning the registration result. Furthermore, we reject pairs that contain the same model point and only keep the pair with the closest point-to-point distance [17]. Together with the simple heuristics from II-D, larger registrations errors resulting from erroneous odometry estimates or laser scans as shown in Figure 1 are typically avoided.

In pose tracking mode, where we match all range scans in a pairwise fashion, we apply an additional pair rejection method called *frustum culling* that was originally designed for the smaller field of view of time-of-flight cameras [18]. The idea is to neglect those points  $\mathbf{d}_{i,j} \in D_i$  in the correspondence search that are not visible from the pose  $\mathbf{x}_{i-1}$  where  $D_{i-1}$  has been taken. In the two-dimensional case and assuming a field-of-view of  $\pi$  (typical for range scans), frustum culling reduces to transforming  $D_i$  into the coordinate frame formed by  $\mathbf{x}_{i-1}$  and removing those points  $\mathbf{d}_{i,j}$  that fall below the  $x$ -axis ( $\mathbf{d}_{i,j}^x < 0$ ). The correspondence search is only conducted for the residual points. It should be noted that the frustum is only a rough approximation of the parts in  $D_i$  that are actually visible from  $\mathbf{x}_{i-1}$  and that frustum culling is only applied in the pairwise registration when performing pose tracking in the presence of odometry errors. However, we are planning to integrate a more sophisticated pair rejection based on visibility into the incremental registration procedure.

### F. Removing Dynamic Points

Up to now points are only added to the incrementally built map but never modified or removed. Hence, range

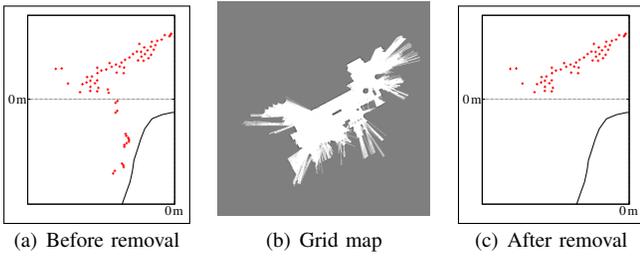


Fig. 2. Removing less probable and dynamic points from a sparse point map. Points measured at the legs of a human passing by are removed from the map in a later registration (data set recorded by Nick Roy).

measurements taken on the surface of dynamic objects cause phantom objects in the resulting point map (see Figure 2). For navigational purposes and later registrations we do not need to explicitly model the corresponding changes. It is sufficient to remove points from regions that are not blocked by static obstacles. For this purpose, we additionally construct a grid map modeling for each cell the probability  $p_{\text{ref}}$  of reflecting a range beam by means of the following simple statistics [19]:

$$p_{\text{ref}}(c^{[xy]}) = \frac{\text{hits}(x,y)}{\text{hits}(x,y) + \text{misses}(x,y)}, \quad (11)$$

where  $\text{hits}(x,y)$  corresponds to the total number of range beams ending up in cell  $c^{[xy]}$  and  $\text{misses}(x,y)$  to the number of beams passing through  $c^{[xy]}$  without being reflected.

In regular intervals it is then checked for every point in  $\mathbf{m}_{i,j} \in M_i$ , whether the reflection probability  $p_{\text{ref}}(c^{[\mathbf{m}_{i,j}]})$  of the corresponding region is larger than some threshold  $p_{\text{min}}$ . Points, where the reflection probability is lower than the threshold, are removed from the point map:

$$\check{M}_i = M_i \setminus \{\mathbf{m}_{i,j} \mid p_{\text{ref}}(c^{[\mathbf{m}_{i,j}]}) < p_{\text{min}}\}. \quad (12)$$

By this means, phantom effects as caused by people passing by are avoided. Since the runtime complexity of the update procedure for grid maps is linear in the number of laser beams and the maximum measurable distance but not on the grid map size, the additional update can easily be carried out after registration without noticeably influencing the overall runtime of the SLAM algorithm. The validity check itself is linear in the number of points stored in the map ( $O(|M|)$ ) since the grid maps allow a direct access to the reflection probability of the corresponding region ( $O(1)$ ).

### III. EVALUATION AND BENCHMARK

Over the last decades, SLAM algorithms have been primarily evaluated by visually inspecting the constructed maps and the estimated trajectory of the robot or by comparing individual registration results as in [7], [8]. Recently, Burgard *et al.* [11] proposed a simple metric for benchmarking SLAM algorithms. It measures the deviations of the determined trajectory from ground truth by evaluating relations between individual poses. These relations represent the geometric relation  $\delta_{i,j} = \mathbf{x}_j \ominus \mathbf{x}_i$  between poses  $\mathbf{x}_i$  and  $\mathbf{x}_j$ . They consist of local relations describing the robot's movement

between consecutive poses and global relations for evaluating the global consistency of the resulting map. Compared are the mean errors in translation  $\bar{\epsilon}_{\text{trans}}$  and rotation  $\bar{\epsilon}_{\text{rot}}$  formed by the deviations of the relations  $\delta_{i,j}$  computed from the determined trajectory and the ground truth relations  $\delta_{i,j}^*$ :

$$\begin{aligned} \epsilon(\delta) &= \frac{1}{N} \sum_{i,j} (\delta_{i,j} \ominus \delta_{i,j}^*)^2 \\ &= \underbrace{\frac{1}{N} \sum_{i,j} \text{trans}(\delta_{i,j} \ominus \delta_{i,j}^*)^2}_{\bar{\epsilon}_{\text{trans}}} + \underbrace{\frac{1}{N} \sum_{i,j} \text{rot}(\delta_{i,j} \ominus \delta_{i,j}^*)^2}_{\bar{\epsilon}_{\text{rot}}}, \end{aligned} \quad (13)$$

where the operators *trans* and *rot* extract, respectively, the translational and rotational part of  $(\delta_{i,j} \ominus \delta_{i,j}^*)$ .

In order to compare the incremental registration algorithm (SPM-ICP) described in Section II with gmapping [4], a widely used and efficient implementation of a Rao-Blackwellized Particle Filter (RBPF), we reproduce the experiments from [11] using the data sets and the ground truth relations available in [20]. The results of this evaluation are summarized in Table I. Presented are both the results reported by Burgard *et al.* in [11] and the results from a direct comparison between the proposed incremental registration and the latest available version of gmapping. For three data sets, the ACES building at the University of Texas in Austin (Figures 3 and 6.a), the INTEL Research Lab (Figures 4 and 6.b) and the CSAIL building at MIT (Figures 5 and 6.c), both the incremental registration as well as gmapping construct consistent environment models showing comparable mean errors in translation and rotation. Only for the ACES data set, there is a minor inconsistency (see Figure 7) in the map of SPM-ICP that is, however, negligible and the robot is able to reliably localize itself in the constructed map.

Unsurprisingly, the incremental registration algorithm is faster than gmapping by a factor of 50 to 100. The measured processing times per scan ( $t_{\text{proc}}$ ) are only meant to provide an approximate trend showing that the proposed registration algorithm can be applied online for matching almost every acquired range scan. This is what makes the pose tracking mode in the presence of larger odometry errors possible at all. In the FR079 data set, for example, the average processing time per scan is  $\bar{t}_{\text{proc}} = 22.156 \pm 5.024$  ms (roughly 45 Hz). For comparison, the SICK LMS 2xx laser scanners deliver range scans with 75 Hz ( $1^\circ$ -resolution) or 37.5 Hz ( $0.5^\circ$ -resolution).

The aforementioned FR079 data set contains larger odometry errors hindering gmapping in constructing globally consistent maps (see Figure 6.d). Here, Burgard *et al.* used a scan matching algorithm in a preprocessing step to correct these errors. With the additional heuristics from Section II-D, the incremental registration algorithm switches into the pose tracking mode when encountering larger jumps and produces a global consistent map without additional preprocessing (see Figure 8).

The MIT Infinite Corridor data set is the only data set in this evaluation for which the incremental registration algo-

TABLE I  
QUANTITATIVE RESULTS FROM BENCHMARKING

	Scanmatching [11]		ICP+SPM (as proposed)		RBPF (50 part.)		Graph Mapping [11]	
	$\bar{\epsilon}_{\text{trans}}/\text{m}$	$\bar{\epsilon}_{\text{rot}}/\text{deg}$	$\bar{\epsilon}_{\text{trans}}/\text{m}$	$\bar{\epsilon}_{\text{rot}}/\text{deg}$	$\bar{\epsilon}_{\text{trans}}/\text{m}$	$\bar{\epsilon}_{\text{rot}}/\text{deg}$	$\bar{\epsilon}_{\text{trans}}/\text{m}$	$\bar{\epsilon}_{\text{rot}}/\text{deg}$
ACES	$0.173 \pm 0.614$	$1.2 \pm 1.5$	$0.060 \pm 0.055$	$1.21 \pm 1.61$	$0.057 \pm 0.046$ <sup>(1)</sup> $0.060 \pm 0.049$ <sup>(2)</sup>	$1.32 \pm 1.74$ <sup>(1)</sup> $1.2 \pm 1.3$ <sup>(2)</sup>	$0.044 \pm 0.044$	$0.4 \pm 0.4$
INTEL	$0.220 \pm 0.296$	$1.7 \pm 4.8$	$0.043 \pm 0.058$	$1.50 \pm 3.07$	$0.067 \pm 0.078$ <sup>(1)</sup> $0.070 \pm 0.083$ <sup>(2)</sup>	$1.85 \pm 2.83$ <sup>(1)</sup> $3.0 \pm 5.3$ <sup>(2)</sup>	$0.031 \pm 0.026$	$1.3 \pm 4.7$
MIT	$1.651 \pm 4.318$	$2.3 \pm 4.5$	$2.283 \pm 6.002$	$1.15 \pm 2.53$	$0.619 \pm 2.048$ <sup>(1)</sup> $0.122 \pm 0.386$ <sup>(2)</sup>	$0.50 \pm 0.61$ <sup>(1)</sup> $0.8 \pm 0.8$ <sup>(2)</sup>	$0.050 \pm 0.056$	$0.5 \pm 0.5$
CSAIL	$0.106 \pm 0.325$	$1.4 \pm 4.5$	$0.043 \pm 0.053$	$1.68 \pm 2.67$	$0.061 \pm 0.129$ <sup>(1)</sup> $0.049 \pm 0.049$ <sup>(2)</sup>	$2.23 \pm 3.09$ <sup>(1)</sup> $0.6 \pm 1.2$ <sup>(2)</sup>	$0.004 \pm 0.009$	$0.05 \pm 0.08$
FR 79	$0.258 \pm 0.427$	$1.7 \pm 2.1$	$0.057 \pm 0.043$	$1.49 \pm 1.71$	$2.709 \pm 5.528$ <sup>(1)</sup> $0.061 \pm 0.044$ <sup>(2)</sup>	$8.35 \pm 12.76$ <sup>(1)</sup> $0.6 \pm 0.6$ <sup>(2)</sup>	$0.056 \pm 0.042$	$0.6 \pm 0.6$

RBPF reference

Proposed approach

(1): with gmapping by Grisetti et al. (<https://svn.openslam.org/data/svn/gmapping, rev. 40>)  
(2): values as provided in [11]  
red (dark gray): resulting maps are not consistent (from visual inspection)

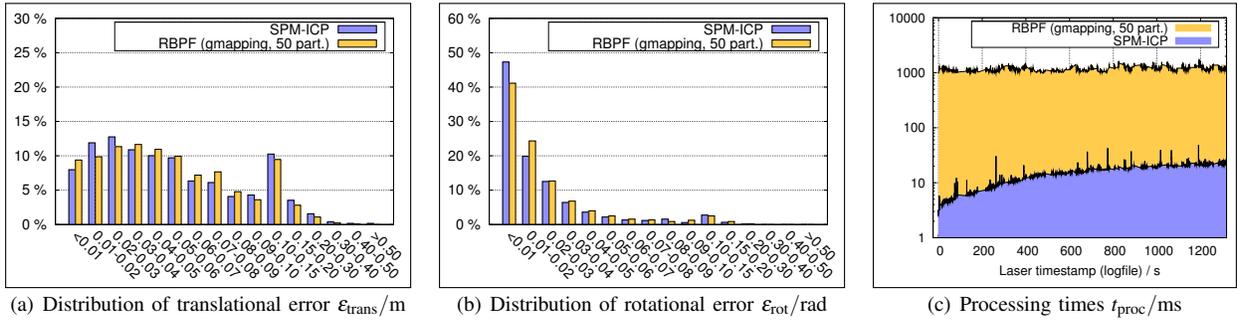


Fig. 3. Error distributions and measured runtimes for the ACES data set (recorded by P. Beeson).

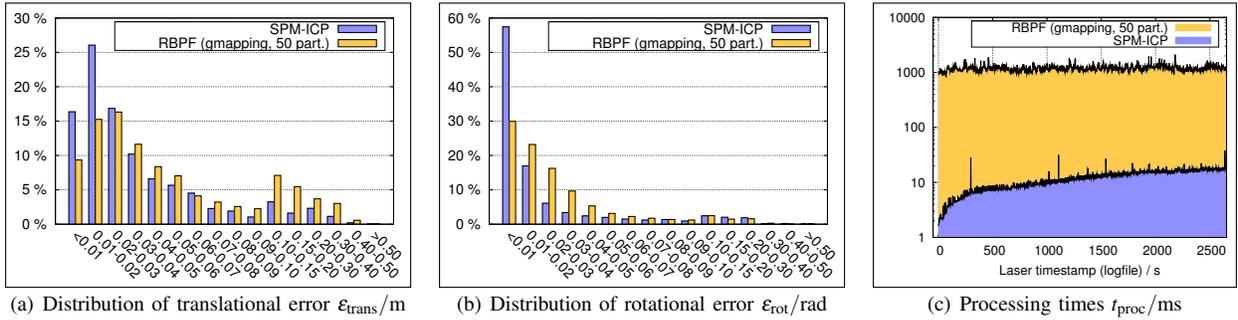


Fig. 4. Error distributions and measured runtimes for the INTEL data set (recorded by D. Hähnel).

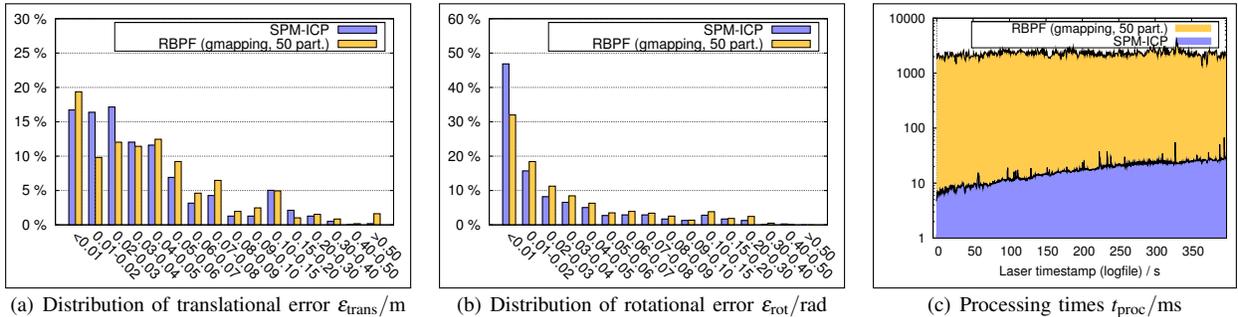


Fig. 5. Error distributions and measured runtimes for the CSAIL data set (recorded by C. Stachniss).

rithm could not construct a globally consistent model. The loops in the robot's trajectory that lead through previously unmodeled terrain are simply too long. Minor registration errors accumulated along a loop lead to deviations between first and last pose on the loop that can no longer be corrected by the incremental registration (see Figure 6.e).

#### IV. CONCLUSION

In this paper, we presented an incremental registration algorithm for SLAM that is based on the ICP algorithm together with a set of extensions for explicitly handling ICP's shortcomings. An exponentially decaying distance threshold between corresponding points together with several heuristics for rejecting correspondence pairs generally improves the registration results. Additional heuristics and switching into matching (almost) every acquired range scan without using odometry in the initial estimates allows for constructing globally consistent maps even in the presence of erroneous jumps in the odometry information. By means of the incremental registration procedure, newly acquired range scans are registered only once against the so far built model and do not need to be kept in memory for later processing. Furthermore, by means of avoiding duplicate point storage in the sparse point map, the memory consumption of the algorithm is low and only increased by additionally constructing and updating a probabilistic reflection grid map. This is used to sort out points belonging to objects that are no longer present in the respective workspace region. Altogether, this simple but efficient approach allows for constructing globally consistent environment models online while being able to process the acquired sensory information in real-time.

In an experimental evaluation, it is shown that the achievable results are comparable with those of the widely used Rao-Blackwellized Particle Filter implementation gmapping being known to build accurate consistent environment models. Especially in smaller environments or those where the robot does not travel for too long through unknown terrain, the proposed algorithm yields accurate and consistent results while not differing in runtime from comparable scan matching algorithms. Because of the latter, it is planned to integrate the proposed registration algorithm into a Rao-Blackwellized Particle Filter in future work. Furthermore, we plan to extend the algorithm to not only estimate the transformation mapping a newly acquired range scan onto the so far built model, but also the corresponding covariance matrix for integrating Graph-SLAM based approaches for distributing accumulated errors.

#### V. ACKNOWLEDGMENTS

Our thanks to C. Stachniss, G. Grisetti, M. Bosse, J. Leonard, D. Hähnel, P. Beeson, N. Roy, J. Minguez, L. Montesano and F. Lamiroux for providing the used data sets as well as W. Burgard, R. Kümmerle, B. Steder, C. Dornhege, M. Ruhnke, G. Grisetti, C. Stachniss and A. Kleiner for providing the ground truth relations and the metric evaluator.

#### REFERENCES

- [1] J. Leonard and H. Feder, "A computationally efficient method for large-scale concurrent mapping and localization," in *International Symposium on Robotics Research*, D. K. J. Hollerbach, Ed., Snowbird, Utah, USA, 1999.
- [2] D. Chekhlov, M. Pupilli, W. Mayol-Cuevas, and A. Calway, "Real-Time and Robust Monocular SLAM Using Predictive Multi-resolution Descriptors," in *Proceedings of the 2nd International Symposium on Visual Computing*, Lake Tahoe, Nevada, USA, 2006, pp. 276–285.
- [3] S. Thrun, Y. Liu, D. Koller, A. Ng, Z. Ghahramani, and H. Durrant-Whyte, "Simultaneous Localization and Mapping with Sparse Extended Information Filters," *International Journal of Robotics Research*, vol. 23, no. 7-8, pp. 693–716, 2004.
- [4] G. Grisetti, C. Stachniss, and W. Burgard, "Improved Techniques for Grid Mapping with Rao-Blackwellized Particle Filters," *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 34–46, 2007.
- [5] E. Olson, J. Leonard, and S. Teller, "Fast Iterative Alignment of Pose Graphs with Poor Estimates," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Orlando, Florida, USA, 2006, pp. 2262–2269.
- [6] G. Grisetti, C. Stachniss, S. Grzonka, and W. Burgard, "A Tree Parameterization for Efficiently Computing Maximum Likelihood Maps using Gradient Descent," in *Proceedings of Robotics: Science and Systems*, Atlanta, GA, USA, 2007.
- [7] A. Nüchter, K. Lingemann, J. Hertzberg, and H. Surmann, "6D SLAM – 3D Mapping Outdoor Environments," *Journal of Field Robotics (JFR), Special Issue on Quantitative Performance Evaluation of Robotic and Intelligent Systems*, vol. 24, no. 8–9, pp. 699–722, 2007.
- [8] M. Magnusson, A. Lilienthal, and T. Duckett, "Scan Registration for Autonomous Mining Vehicles Using 3D-NDT," *Journal of Field Robotics*, vol. 24, no. 10, pp. 803–827, 2007.
- [9] K. Pulli, "Multiview Registration for Large Data Sets," in *Proceedings of the International Conference on 3D Digital Imaging and Modeling*, Ottawa, Canada, 1999, pp. 160–168.
- [10] Y. Chen and G. Medioni, "Object modelling by registration of multiple range images," *Image and Vision Computing, Special Issue on Range Image Understanding*, vol. 10, no. 3, pp. 145–155, 1992.
- [11] W. Burgard, C. Stachniss, G. Grisetti, B. Steder, R. Kuemmerle, C. Dornhege, M. Ruhnke, A. Kleiner, and J. D. Tardos, "A Comparison of SLAM Algorithms Based on a Graph of Relations," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, St. Louis, Missouri, USA, 2009, pp. 2089–2095.
- [12] S. Rusinkiewicz and M. Levoy, "Efficient Variants of the ICP Algorithm," in *Proceedings of the International Conference on 3D Digital Imaging and Modeling (3DIM)*, Quebec, Canada, 2001, pp. 145–152.
- [13] P. J. Besl and N. D. McKay, "A Method for Registration of 3-D Shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, 1992.
- [14] B. K. P. Horn, "Closed-form solution of absolute orientation using unit quaternions," *Journal of the Optical Society of America A*, vol. 4, no. 4, pp. 629–642, 1987.
- [15] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Y. Wu, "An optimal algorithm for approximate nearest neighbor searching in fixed dimensions," *Journal of the ACM (JACM)*, vol. 45, no. 6, pp. 891–923, 1998.
- [16] J. Minguez, "Metric-Based Scan Matching Algorithms for Mobile Robot Displacement Estimation," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2005.
- [17] T. Zinßer, J. Schmidt, and H. Niemann, "A Refined ICP Algorithm for Robust 3-D Correspondence Estimation," in *Proceedings of the International Conference on Image Processing (ICIP)*, Barcelona, Spain, 2003, pp. 695–698.
- [18] S. May, D. Droschel, D. Holz, S. Fuchs, and A. Nüchter, "Robust 3D-Mapping with Time-of-Flight Cameras," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, St. Louis, Missouri, USA, 2009, pp. 1673–1678.
- [19] M. Bennewitz, C. Stachniss, S. Behnke, and W. Burgard, "Utilizing Reflection Properties of Surfaces to Improve Mobile Robot Localization," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Kobe, Japan, 2009, pp. 4287–4292.
- [20] R. Kümmerle, B. Steder, C. Dornhege, M. Ruhnke, G. Grisetti, C. Stachniss, and A. Kleiner, "Slam benchmarking webpage," 2009. [Online]. Available: <http://ais.informatik.uni-freiburg.de/slamevaluation>

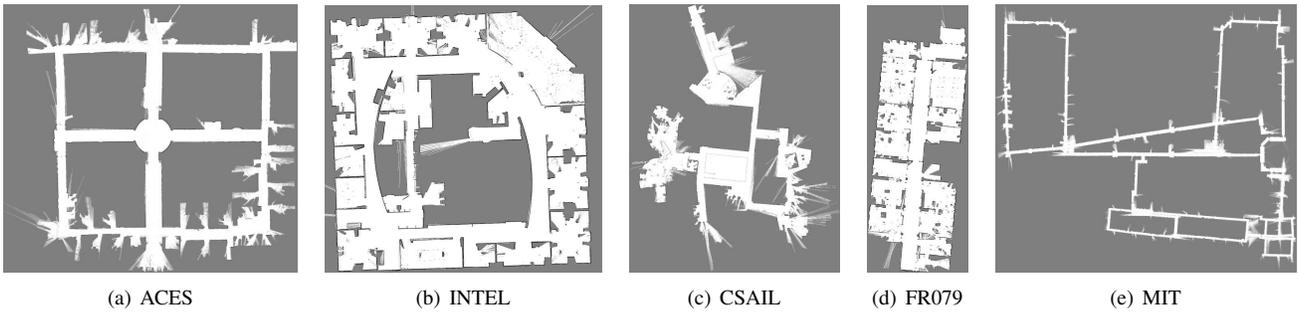


Fig. 6. Constructed grid maps for the five data sets. Except for the MIT data set, the incremental registration algorithm constructed globally consistent maps.

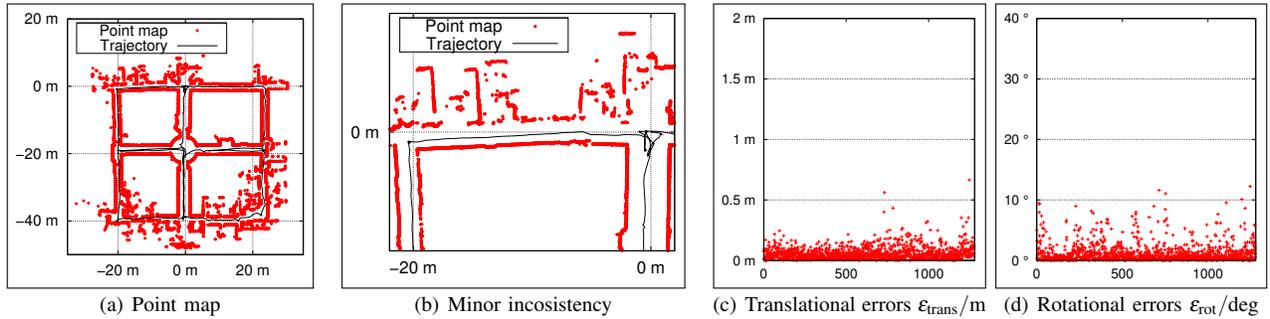


Fig. 7. Minor inconsistency in the map constructed from the ACES data set. The incremental registration induces small errors in relations between consecutive poses to recover from accumulated errors. The result is a globally almost consistent map with only a minor inconsistency not affecting localization or other navigational tasks. The inconsistency (corridors not being parallel/perpendicular) arises in the top left corner of (b) when closing the first loop. These errors are visible in (c) and (d) in the right most data points.

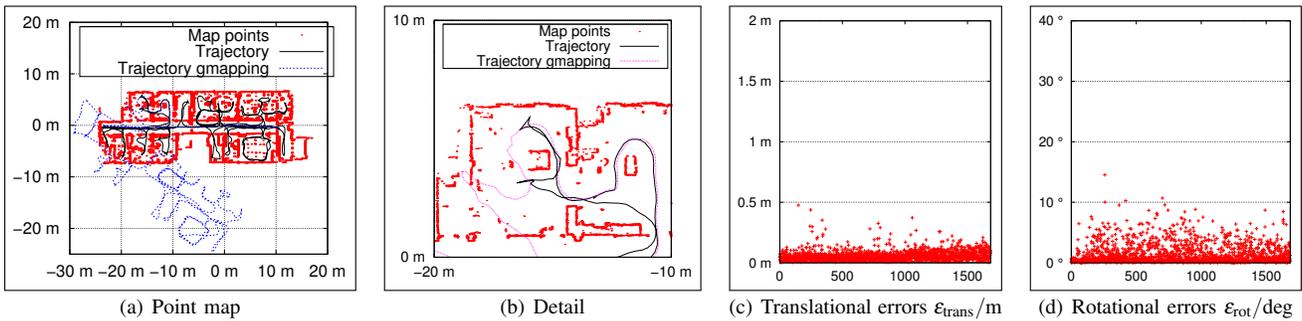


Fig. 8. Constructing consistent maps in the presence of large odometry errors. The odometric pose estimates in the FR079 data set contain several larger jumps hindering gmapping from reproducing the correct trajectory. In the incremental registration algorithm, the additional heuristics enforce matching every range scan (neglecting odometry information), in contrast to only matching a range scan after moving or turning for a certain while, and allow for constructing an accurate globally consistent map.

The used data sets as well as the written log files for reproducing the presented experiments (including additional material and videos) are available at <http://purl.org/holz/spmicp>.