# Online SLAM Based on a Fast Scan-Matching Algorithm

Eurico Pedrosa, Nuno Lau, and Artur Pereira

University Of Aveiro,
Institute of Electronics and Telematics Engineering of Aveiro, Portugal
{efp,nunolau,artur}@ua.pt
http://www.ieeta.pt

**Abstract.** This paper presents a scan-matching approach for online simultaneous localization and mapping. This approach combines a fast and efficient scan-matching algorithm for localization with dynamic and approximate likelihood fields to incrementally build a map. The achievable results of the approach are evaluated using an objective benchmark designed to compare SLAM solutions that use different methods. The result is a fast online SLAM approach suitable for real-time operations.

**Keywords:** scan-matching, localization, mapping, simultaneous localization and mapping (SLAM), real-time.

## 1 Introduction

A basic requirement of an autonomous service robot is the capability to selflocalize in the real-world indoor and domestic environment where it operates. These type of environments are typically dynamic, cluttered and populated by humans (e.g. Figure 1). Not only should the robot act adequately and in due time to the perceived dynamic changes in the environment, it should also be robust to inconsistencies between the collected sensorial information and its internal world representation. The latter is of utmost importance for localization in dynamic environments like the ones already mentioned.

Providing the robot with the required representation of the environment (i.e. obstacles, free space and unknown space) can be a cumbersome task, even unfeasible, when done manually. Simultaneous Localization and Mapping (SLAM) provides the means towards a truly autonomous robot, by minimizing manual input. A robot carrying out SLAM has, in principle, the capability to autonomously construct an internal representation (or map) of the environment. The resulting map captures the structure of the environment that have been sensed at least once.

A common approach to SLAM in real-world applications is to manually direct the robot through the environment while collecting the acquired sensory information. The collected data is then processed offline to construct a representation of the world that can be used by the robot for localization in subsequent tasks. However, there are several tasks that require the robot to construct its internal

L. Correia, L.P. Reis, and J. Cascalho (Eds.): EPIA 2013, LNAI 8154, pp. 295–306, 2013.

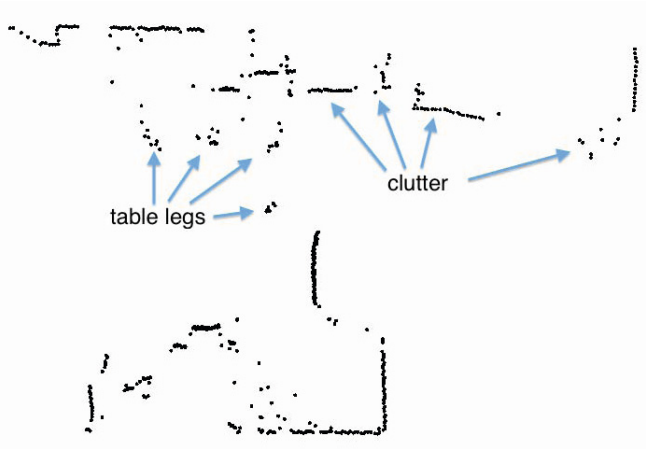



**Fig. 1.** Example of scan acquired in an office space

representation of the environment during operation, like, for instance, following a person through unknown space. The applied algorithm needs to be efficient so that it can be used during operation and to allow for further sensor data processing in real-time.

In this paper, we present a fast and practical online SLAM algorithm suitable for real-time usage. The overall algorithm is based on a fast, scan-matching based, localization algorithm for mobile robots. The remainder of this paper is organized as follow: in the next section related work is presented; in section 3 the localization algorithm is described; in section 4 the necessary changes and additions to the base localization algorithm for performing SLAM are detailed; section 5 presents practical experiments and adequate evaluation of the proposed SLAM algorithm; finally, section 6 presents the conclusion of this paper.

## 2 Related Work

There has been a fair amount of research into the SLAM problem in the last two decades. The probabilistic SLAM trend provide us with solutions that explicitly handle uncertainties about pose estimations and sensory information by estimating a probability distribution over the potential solutions. Example of such approaches are the Extended Kalman Filters (EKFs) [17], Unscented Kalman Filters (UKFs) [3], Sparse Extended Information Filters (SEIFs) [25], and Rao-Blackwellized Particle Filters (RBPFs) [9]. Notwithstanding their robust and accurate results, the necessary computational effort makes it difficult to use the previous techniques for real-time applicability.

SLAM algorithms based on graphs [10,21] compose the problem in terms of a graph where nodes are poses of the robot during mapping and edges are spacial constraint between the nodes. They are typically data association problem solvers with highly efficient algorithms used to find the most probable configuration of nodes given their constraints.

Iterative Closest Point (ICP) [26] and Iterative Closest Line (ICL) (e.g. [2]) are widely used in scan-matching. They both try to find the rigid-body transformation that best align the reference and the query points. Lu and Milios [18] describe methods using ICP and LIDAR scans for localization and map building. Peter Biber proposed the Normal Distribution Transform (NDT) [1] for SLAM. Using a grid map, a normal distribution is assigned to each cell which locally models the probability of measuring a point. This representation is then used for localization, making use of a gradient descent algorithm, and incrementally building a map. Holz *et al.* [11] proposed a fast ICP-based method for SLAM. It is based on range image registration using ICP and sparse point maps. Kohlbrecher *et al.* [12] proposed the combination of scan-matching using a LIDAR system with a 3D attitude estimation. A multi-resolution occupancy grid maps and approximation of map gradients are used for 2D localization and mapping.

## 3 The Perfect Match Algorithm

The original Perfect Match (PM) algorithm was developed by Lauer *et al.* [16] to provide a robust real time self-localization, in a highly dynamic but structured environment, to a mobile robot using a camera system to sense the world.

To the best of our knowledge, an initial adaptation of PM to indoor localization was proposed by Gouveia *et al.* [8] using range data from a LIDAR. Cunha *et al.* [4] proposes the use of PM using depth information. Both follow the original implementation of the algorithm with hand built maps that are not practical for a general use because of the geometric diversity of indoor environments. Here, we assume that the provided map was built in an automatic way, e.g. with SLAM. This can be seen as a prelude for map building using PM.

Without loosing the original properties of the algorithm, PM will be presented within a probabilistic formulation, a trending and well grounded point of view [24]. Furthermore, global localization and pose tracking will be omitted, for they are not relevant to this paper.

### 3.1 Probabilistic Formulation

Let $s_t$ be the robot state at time $t$, furthermore, let $z_t$ be the measurement at time $t$. The control $u_t$ at time $t$ determines the change of state in the time interval $(t-1, t]$, and the map $m$ models the environment. The posterior probability $p(s_t|z_{1:t}, u_{1:t}, m)$ is given by the following recursive update equation [23]:

$$p(s_t|z_{1:t}, u_{1:t}, m) = \eta\, p(z_t|s_t, m) \int \begin{array}{l} p(s_t|u_t, s_{t-1}, m) \\ p(s_{t-1}|z_{1:t-1}, u_{1:t-1}, m) ds_{t-1} \end{array}, \tag{1}$$

where $\eta$ is a normalizer. Furthermore, assuming that the robot motion does not depend of the map $m$, and assuming that $s_{t-1}$ is known at time $t-1$, the posterior for localization is:

$$p(s_t|z_{1:t}, u_{1:t}, m) \propto (p(z_t|s_t, m)p(s_t|\hat{s}_{t-1}, u_t)) \ . \tag{2}$$

The first term, $p(z_t|s_t, m)$, is the observation model, and the second term, $p(s_t|u_t, s_{t-1})$, is the motion model.

The motion model $p(s_t|u_t, s_{t-1})$ describes the probability of the robot pose $s_t$ after executing the control $u_t$, that is, a probabilistic generalization of the robot kinematics. This model is typically known in terms of a multivariate Gaussian.

The measurement model $p(z_t|s_t, m)$, physically models the sensors of the robot. Assuming that the robot pose $s_t$ and the map $m$ are known, the measurements model specifies the probability that $z_t$ is measured. It is common, for sensors equipped in robots, to generates more than one measurement at time $t$. Thus, the result calculation is the collection of probabilities $p(z_t^i|s_t, m)$, where $z_t^i$ is the $i$th measurement. Assuming conditional independence between the measurements, the resulting probability is given by:

$$p(z_t|s_t, m) = \prod_i p(z_t^i|s_t, m) \ . \tag{3}$$

### 3.2 Likelihood Field for Range Finders

The inception of PM has assumed scans extracted from processed images [19], in a way it mimics a range finder (e.g. sonar or LIDAR). Therefore, using range data from a LIDAR, a more precise sensor, is a change without cost.

The adopted model for measurement is the *likelihood field* [23]. It is a model that lacks a plausible physical explanation, like the beam model [5]. However, it works well in practice and its computation is more efficient.

While the *beam model* applies a *ray casting* function to find the "true" $z_t^{i*}$ range of the object measured by $z_t^i$, the *likelihood field* projects the endpoint of $z_t^i$ into global coordinates of the map $m$ to find the distance to the nearest object. Let $s_t = (x\ y\ \theta)^T$ denote the robot pose at time $t$, $(x_{i,\text{sens}}\ y_{i,\text{sens}})^T$ the relative position of the sensor on frame of the robot, and $\theta_{i,\text{sens}}$ the angular orientation of the sensor relative to the heading of the robot. The endpoint of the measurement $z_t^i$, in global coordinates, is given by the transformation $\mathcal{T}$:

$$\begin{pmatrix} x_{z_t^i} \\ y_{z_t^i} \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \left[ \begin{pmatrix} x_{i,\text{sens}} \\ y_{i,\text{sens}} \end{pmatrix} + z_t^i \begin{pmatrix} \cos\theta_{i,\text{sens}} \\ \sin\theta_{i,\text{sens}} \end{pmatrix} \right] \tag{4}$$

The noise of the process is modeled by a probability density function (pdf) that requires finding the nearest object in the map. Let $\Delta_i$ denote the Euclidean distance between $(x_{z_t^i}\ y_{z_t^i})^T$ and the nearest object in the map $m$. Then, measurement noise can be modeled by

$$p(z_t^i|s_t, m) = \eta_i \frac{\sigma^2}{\sigma^2 + \Delta_i^2} \ , \tag{5}$$

a derivation of the error function presented in the original PM [16]. Furthermore, the value measured by a range sensor is, in practice, bounded by the interval

$[0, z_{\max}]$, where $z_{\max}$ is the maximum value that a measurement yields. Hence, the normalizer $\eta$ is given by

$$\eta_i = \left( \int_0^{z_{\max}} \frac{\sigma^2}{\sigma^2 + \Delta_i^2} \, dz_t^i \right)^{-1} . \tag{6}$$

The value of $\Delta_i$ is computed by a search function defined by

$$\mathcal{D}(x_{z_t^i}, y_{z_t^i}) = \min_{x',y'} \left( \sqrt{(x_{z_t^i} - x')^2 + (y_{z_t^i} - y')^2} \mid \langle x', y' \rangle \text{occupied in } m \right) . \tag{7}$$

To speed up this search, $\mathcal{D}$ is a look-up table created by computing the Euclidean distance transform of the map. First, the map is converted to a binary occupancy map, then a simple and fast method is used to compute the distance transformation [7]. This method operates over discrete grids, therefore, the obtained result is only an approximation. To improve this approximation an interpolation scheme, using bilinear filtering, is employed to estimate a more accurate $\Delta_i$ [12].

### 3.3 Maximum Likelihood Pose Estimation

The *maximum likelihood* approach for pose estimation, although non-probabilistic, is simpler and easier to calculate than the posterior (1). The idea is simple: given a measurement and odometry reading, calculate the most likely pose. Mathematically speaking, the $s_t$ pose is obtained as the maximum likelihood estimate of (2):

$$\hat{s}_t = \arg\max_{s_t} \; p(z_t|s_t, m) \, p(s_t|\hat{s}_{t-1}, u_t) \tag{8}$$

To summarize, in time $t-1$ the (non-probabilistic) estimate of $\hat{s}_{t-1}$ is given to the robot. As $u_t$ is executed and a new $z_t$ is obtained, the most likely pose $\hat{s}_t$ is calculated by the robot.

In this approach $\hat{s}_t$ is found using a gradient ascent algorithm in log likelihood space. It is common to maximize the log likelihood instead of the likelihood because it is mathematically easier to handle, and the maximization is justified by the fact that it is a strict monotonic function. Thus, this approach tries to calculate

$$\hat{s}_t = \arg\max_{s_t} \; \ln[p(z_t|s_t, m) \, p(s_t|\hat{s}_{t-1}, u_t)] . \tag{9}$$

Taking advantage of the properties of the logarithm, this expression can be decomposed into additive terms:

$$\hat{s}_t = \arg\max_{s_t} \; \ln \, p(z_t|s_t, m) + \ln \, p(s_t|\hat{s}_{t-1}, u_t) . \tag{10}$$

The required differentiation with respect to the pose $s_t$ are:

$$\nabla_{s_t} L = \nabla_{s_t} \ln \, p(z_t|s_t, m) + \nabla \ln \, p(s_t|\hat{s}_{t-1}, u_t) , \tag{11}$$

where $L$ is the log likelihood. To simplify the calculation of $\nabla_{s_t} L$ the motion model, in this approach, is assumed to have constant probability, therefore the

calculation of $\nabla \ln p(s_t|\hat{s}_{t-1}, u_t)$ is not required. Consequently, the required gradient is given by:

$$\begin{aligned}\nabla_{s_t} L =& \nabla_{s_t} |z_t| \ln \eta + 2|z_t| \ln \sigma - \sum_i \ln(\sigma^2 + \Delta^2) \\ =& -\sum_i \frac{2\Delta}{\sigma^2 + \Delta^2} \nabla_{\mathcal{T}} \mathcal{D}(x_{z_t^i}, x_{z_t^i})^T \begin{bmatrix} 1 & 0 & (-\hat{x}_{z_t^i} \sin\theta - \hat{y}_{z_t^i} \cos\theta) \\ 0 & 1 & (\hat{x}_{z_t^i} \cos\theta - \hat{y}_{z_t^i} \sin\theta) \end{bmatrix} ,\end{aligned} \tag{12}$$

where $(\hat{x}_{z_t^i} \ \hat{y}_{z_t^i})^T$ is the $z_t^i$ endpoint relative to the robot, and $\nabla_{\mathcal{T}} \mathcal{D}(x_{z_t^i}, x_{z_t^i})$ is the gradient of the Euclidean *distance transform* with respect to $\mathcal{T}$ (see (4) ). This gradient can not be presented in closed form, but is calculated with the *sobel* operator.

The gradient ascent maximizes the log likelihood interactively changing the pose $s_t$ in the direction of the gradient. The aforementioned maximization is computed with the RPROP algorithm [22], which is capable of providing practical result in 10 iterations.

## 4 Mapping with Perfect Match

Mapping is the problem of generating a map from measurement values, which can be easy to execute if done with known locations [23]. The localization problem with a map is also relatively simple. However when both are combined the problem has a much higher dimensionality, one could argue that the problem has infinite dimensions [24].

The PM is a fast and robust algorithm for localization based on scan-matching. This type of algorithm has already been augmented for SLAM (e.g. [18,12]). Thus, knowing that PM is fast (i.e. execution times of about $1ms$ in our experiments), the goal is to create an also fast and practical online SLAM solution using PM.

### 4.1 Concurrent Mapping and Localization

To construct the map, a function for incrementally building maps with knowledge of the poses of the robot is required. The function $\hat{m}(s_{1:t}, z_{1:t})$ is used to incrementally build the map as a probabilistic occupancy grid [6,20].

Let us follow the same probabilistic framework presented for localization. By augmenting the state $s$ that is being estimated by the map $m$, and maintaining the same assumptions already made, the posterior can be defined by:

$$p(s_t, \hat{m}(s_{1:t}, z_{1:t})|z_{1:t}, u_{1:t}) \propto p(z_t|s_t, \hat{m}(s_{1:t}, z_{1:t}))p(s_t|\hat{s}_{t-1}, u_t) \ . \tag{13}$$

The change of $m$ by $\hat{m}(s_{1:t}, z_{1:t})$ also propagates to the estimation of the robot pose $\hat{s}_t$ (see subsection 3.3). The implication of this formulation, is that PM is used for localization using the map constructed so far, i.e. $\hat{m}(s_{1:t-1}, z_{1:t-1})$.

The initial distribution $p(s, m)$ is not known, but it is straightforward. The initial pose is set to $s = \langle 0, 0, 0 \rangle$, and the occupancy grid that models the map

is initialized with an uniform prior. However, by having an uniform map the localization would fail at the very first execution due to the lack of a reference. Therefore, the first estimation of the robot pose $\hat{s}$ is equal to the initial $s$.

### 4.2 Dynamic Likelihood Field

PM requires a known map to work, and its speed relies in the use of the *likelihood field* model and the RPROP algorithm. In essence, the computation of the robot pose $s_t$ with PM is the result of several transformations $\mathcal{T}$ for table look-ups (*likelihood field*) during the RPROP iterations to find a local maximum. The created tables are static over time, therefore, the overhead time introduced by their creation is negligible because it happens only once.

In the presented approach the map is built incrementally, thus as a result, the map changes over time and the static assumption falls through. For the PM to work, each time the map changes the Euclidean distance map $\mathcal{D}$ and gradient must be recalculated. For small maps their computation time can be ignored, however, as the map grows in dimension PM looses its *fast* and *real-time* "badges".

To solve the execution time degradation a dynamic update of the Euclidean distance map is employed. Let $\hat{\mathcal{D}}$ denote the Euclidean distance map with relation to $\hat{m}(s_{1:t}, z_{1:t})$. The implemented algorithm was presented by Lau *et al.* [15], and it seek to update only the cells affected by the change of state (i.e. from occupied to free or vice versa) of the cells involved in the update of the map. Additionally, the same interpolation scheme used in section 3.2 is employed.

The gradient $\nabla_{\mathcal{T}}\hat{\mathcal{D}}(x_{z_t^i}, y_{z_t^i})$ also needs to be calculated. Applying a *sobel* operator each time $\hat{\mathcal{D}}$ changes creates the same kind of temporal overhead already discussed for updating $\mathcal{D}$. The adopted solution is the one presented in [12], where the gradient is an interpolation of the difference of the four closest samples in the map along $x$- and $y$-axis.

## 5 Experiments and Evaluation

The initial experiments were carried out on an office type environment dataset with a fair amount of clutter, chairs and tables, and people walking. The legs of the chairs and tables create a considerable amount of misreadings from the LIDAR. It has a long corridor, but it lacks a loop. The obtained map was visually compared with a map generated with GMapping (see Figure 2), a state-of-the-art RBPF implementation [9], and both are very similar, meaning that our proposal can achieve practical results. However since it is a subjective evaluation, further tests were done.

Kümmerle *et al.* [14] proposes an objective benchmark for evaluating SLAM solutions. Instead of using the map itself in the evaluation procedure, the poses of the robot during data acquisition are used. The benchmark allows to compare different algorithm independently of the type of map representation they use,

**(a)** Map built with GMapping.

**(b)** Map built with the proposed approach.

**Fig. 2.** Visual comparison of the same map built from the same dataset but with different SLAM approaches

such as occupancy grids or feature maps. The metric is based on relative displacement between robot poses $\delta_{i,j} = s_j \ominus s_i$ instead of comparing poses in the global coordinates. The comparison is made with the mean error in translation $\bar{\epsilon}_{\text{trans}}$ and mean error in rotation $\bar{\epsilon}_{\text{rot}}$ calculated from the obtained trajectory relation $\delta_{i,j}$ and the ground truth relation $\delta^*_{i,j}$:

$$\epsilon(\delta) = \underbrace{\frac{1}{N}\sum_{i,j} \text{trans}(\delta_{i,j} \ominus \delta^*_{i,j})^2}_{\bar{\epsilon}_{\text{trans}}} + \underbrace{\frac{1}{N}\sum_{i,j} \text{rot}(\delta_{i,j} \ominus \delta^*_{i,j})^2}_{\bar{\epsilon}_{\text{rot}}} , \tag{14}$$

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

To compare the PM for map building described in section 4 with other SLAM approaches, selected experiences from [14] were repeated using the datasets and evaluation software available in [13]. The obtained results are summarized in Table 1 and Table 2. The presented results also include the ones reported by Kümmerle *et al.* [14] and Holz *et al.* [11].

For each one of the selected datasets, the ACES building at the University of Texas in Austin (Figure 3a), the Intel Research Lab (Figure 3b), and the Freiburg Building 079 (Figure 3c), a consistent map was built using the proposed approach. The translational and rotational errors are, in the majority of the cases, lower than those from the other approaches. Additionally, an approximate distribution of the errors is visually presented in Figure 4 and Figure 5.

As expected, the PM algorithm for map building is also fast. The measured execution time per scan ($\bar{t}_{\text{scan}}$) provides an approximate trend (see Figure 6) showing that the proposed method can be applied online for most, if not all,

**Table 1.** Quantitative results of different datasets/approaches on the absolute translational error

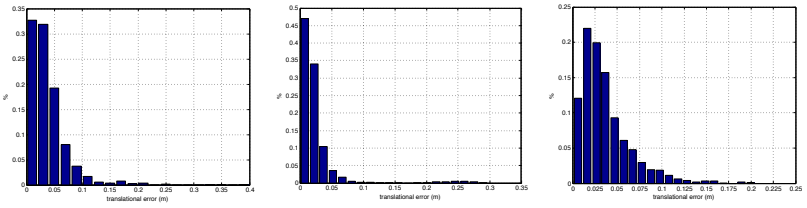
| | PM (proposed) $\bar{\epsilon}_{\text{trans}}/m$ | ICP [11] $\bar{\epsilon}_{\text{trans}}/m$ | Scanmatching [14] $\bar{\epsilon}_{\text{trans}}/m$ | RBPF [14] $\bar{\epsilon}_{\text{trans}}/m$ | GraphMapping [14] $\bar{\epsilon}_{\text{trans}}/m$ |
|---|---|---|---|---|---|
| ACES | $0.038 \pm 0.036$ | $0.060 \pm 0.055$ | $0.173 \pm 0.614$ | $0.060 \pm 0.049$ | $0.044 \pm 0.044$ |
| INTEL | $0.025 \pm 0.037$ | $0.043 \pm 0.058$ | $0.220 \pm 0.296$ | $0.070 \pm 0.083$ | $0.031 \pm 0.026$ |
| FR79 | $0.037 \pm 0.028$ | $0.057 \pm 0.043$ | $0.258 \pm 0.427$ | $0.061 \pm 0.044$ | $0.056 \pm 0.042$ |

**Table 2.** Quantitative results of different datasets/approaches on the absolute rotational error

| | PM (proposed) $\bar{\epsilon}_{\text{rot}}/def$ | ICP [11] $\bar{\epsilon}_{\text{rot}}/deg$ | Scanmatching [14] $\bar{\epsilon}_{\text{rot}}/deg$ | RBPF [14] $\bar{\epsilon}_{\text{rot}}/deg$ | GraphMapping [14] $\bar{\epsilon}_{\text{rot}}/deg$ |
|---|---|---|---|---|---|
| ACES | $0.55 \pm 1.15$ | $1.21 \pm 1.61$ | $1.2 \pm 1.5$ | $1.2 \pm 1.3$ | $0.4 \pm 0.4$ |
| INTEL | $1.03 \pm 2.92$ | $1.50 \pm 3.07$ | $1.7 \pm 4.8$ | $3.0 \pm 5.3$ | $1.3 \pm 4.7$ |
| FR79 | $0.51 \pm 0.74$ | $1.49 \pm 1.71$ | $1.7 \pm 2.1$ | $0.6 \pm 0.6$ | $0.6 \pm 0.6$ |

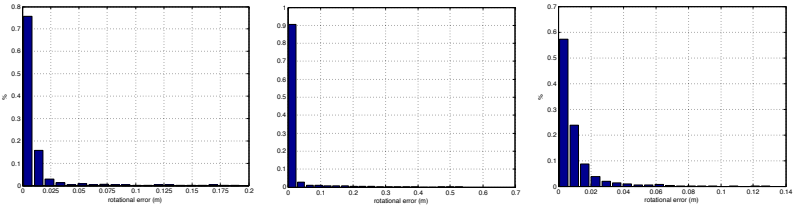
**(a)** ACES **(b)** Intel **(c)** FR79

**Fig. 3.** Constructed occupancy grid maps for the three datasets using the proposed approach. From visual inspection all maps all globally consistent.



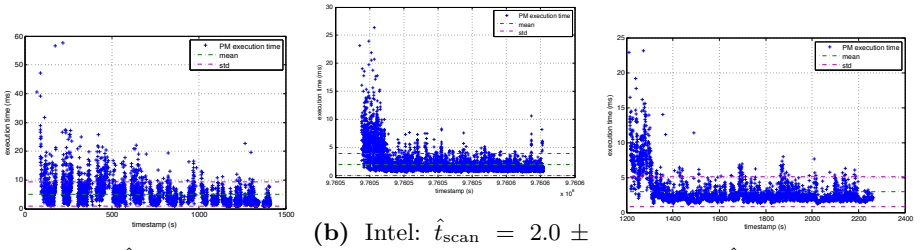

**(a)** ACES **(b)** Intel **(c)** FR79

**Fig. 4.** Distribution of the translational errors of the proposed approach for the three datasets




**(a)** ACES **(b)** Intel **(c)** FR79

**Fig. 5.** Distribution of the rotational errors of the proposed approach for the three datasets




**(a)** ACES: $\hat{t}_{\text{scan}} = 5.1 \pm 4.2s$ **(b)** Intel: $\hat{t}_{\text{scan}} = 2.0 \pm 2.0s$ **(c)** FR79: $\hat{t}_{\text{scan}} = 3.0 \pm 2.1s$

**Fig. 6.** Execution times of the proposed approach for the three datasets.

scans. The value of $\bar{t}_{\text{scan}}$ has a tendency to decrease as more information about the environment is added to the map. This can be explained by the fact that revised areas of the environment introduce smaller changes to the map, resulting in smaller updates to the Euclidean distance map $\hat{\mathcal{D}}$. There are other factors that can influence $\bar{t}_{\text{scan}}$ , including the number of measurements $z_t^i$ and their range, the number of iterations performed by RPROP, the granularity of the map $\hat{m}$ , and obviously the hardware used for computation. In the conducted experiments, a MacBook Pro with an Intel Core i7 2.8GHz and 4GB of RAM was used for the construction of the maps with $0.05m$ of resolution updated with measurements $z_t$ with a maximum range of $40m$ after 30 optimizations of RPROP.

It should also be noticed that no pre-processing was applied to the used datasets, which was not true in the experiments of Kümmerle *et al.* [14] and Holz *et al.* [11]. Instead, to cope with the unreliable odometry of the datasets, shorter accumulated displacement, $0.01m$ for translation and $0.5rad$ for rotation, was used as threshold for update. As result, smaller errors had to be handle. For comparison, in the office dataset, an accumulated displacement of $1.0m$ for translation and $0.5rad$ for rotation was used.

## 6 Conclusion

In this paper an incremental map building algorithm, based on the scan-matching algorithm PM, was presented. Several modification were made to the original

algorithm, including the use of a dynamic *likelihood field* capable of a fast adaptation to the changes of the map being builr, while maintaining the necessary properties for the localization part of SLAM.

In the experimental evaluation, it is shown that the obtained results are comparable with those of the extensively used GMapping, and others, to construct a suitable model of the perceived environment. Furthermore, the resulting execution times show that the method in question has run-time characteristics that makes it practical for concurrent operations with real-time restrictions.

**Acknowledgments.** This research is supported by: FEDER through the Operational Program Competitiveness Factors - COMPETE and by National Funds through FCT in the context of the project FCOMP-01-0124-FEDER-022682 (FCT reference PEst-C/EEI/UI0127/2011); the project Cloud Thinking (funded by the QREN Mais Centro program, ref. CENTRO-07-ST24-FEDER-002031); and PRODUTECH-PTI, QREN 13851, by National Funds through FCT/MCTES (PIDDAC) by FEDER through COMPETE – POFC.

## References

1. Biber, P., Strasser, W.: The normal distributions transform: a new approach to laser scan matching. In: Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003), vol. 3, pp. 2743–2748 (2003)
2. Censi, A.: An ICP variant using a point-to-line metric. In: 2011 IEEE International Conference on Robotics and Automation (ICRA), pp. 19–25. IEEE (2008)
3. Chekhlov, D., Pupilli, M., Mayol-Cuevas, W., Calway, A.: Real-time and robust monocular slam using predictive multi-resolution descriptors. In: Bebis, G., et al. (eds.) ISVC 2006, Part II. LNCS, vol. 4292, pp. 276–285. Springer, Heidelberg (2006), http://dx.doi.org/10.1007/11919629_29
4. Cunha, J., Pedrosa, E., Cruz, C., Neves, A.J., Lau, N.: Using a depth camera for indoor robot localization and navigation. In: RGB-D: Advanced Reasoning with Depth Cameras - RSS Workshop (2011), http://www.cs.washington.edu/ai/Mobile_Robotics/rgbd-workshop-2011
5. Dellaert, F., Fox, D., Burgard, W., Thrun, S.: Monte carlo localization for mobile robots. In: Proceedings of the 1999 IEEE International Conference on Robotics and Automation, vol. 2, pp. 1322–1328 (1999)
6. Elfes, A.: Occupancy grids: a probabilistic framework for robot perception and navigation. Ph.D. thesis, Carnegie Mellon University (1989)
7. Felzenszwalb, P., Huttenlocher, D.: Distance Transforms of Sampled Functions. Tech. rep., Cornell University (2004)
8. Gouveia, M., Moreira, A.P., Costa, P., Reis, L.P., Ferreira, M.: Robustness and precision analysis in map-matching based mobile robot self-localization. In: New Trends in Artificial Intelligence: 14th Portuguese Conference on Artificial Intelligence, pp. 243–253 (2009)
9. Grisetti, G., Stachniss, C., Burgard, W.: Improved Techniques for Grid Mapping With Rao-Blackwellized Particle Filters. IEEE Transactions on Robotics 23(1), 34–46 (2007)
10. Grisetti, G., Stachniss, C., Grzonka, S., Burgard, W.: A tree parameterization for efficiently computing maximum likelihood maps using gradient descent. In: Proc. of Robotics: Science and Systems, RSS (2007)

11. Holz, D., Behnke, S.: Sancta simplicitas - on the efficiency and achievable results of SLAM using ICP-based incremental registration. In: 2011 IEEE International Conference on Robotics and Automation (ICRA), pp. 1380–1387 (2010)
12. Kohlbrecher, S., von Stryk, O., Meyer, J., Klingauf, U.: A flexible and scalable SLAM system with full 3D motion estimation. In: 2011 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR), pp. 155–160. IEEE (2011)
13. Kümmerle, R., Steder, B., Dornhege, C., Ruhnke, M., Grisetti, G., Stachniss, C., Kleine, A.: SLAM benchmarking (2009), `http://kaspar.informatik.uni-freiburg.de/~slamEvaluation/datasets.php`
14. Kümmerle, R., Steder, B., Dornhege, C., Ruhnke, M., Grisetti, G., Stachniss, C., Kleiner, A.: On measuring the accuracy of SLAM algorithms. Autonomous Robots 27(4), 387–407 (2009)
15. Lau, B., Sprunk, C., Burgard, W.: Improved updating of Euclidean distance maps and Voronoi diagrams. In: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 281–286 (2010)
16. Lauer, M., Lange, S., Riedmiller, M.: Calculating the perfect match: An efficient and accurate approach for robot self-localization. In: Bredenfeld, A., Jacoff, A., Noda, I., Takahashi, Y. (eds.) RoboCup 2005. LNCS (LNAI), vol. 4020, pp. 142–153. Springer, Heidelberg (2006)
17. Leonard, J.J., Feder, H.J.S.: A Computationally Efficient Method for Large-Scale Concurrent Mapping and Localization. In: International Symposium of Robotics Research (2000)
18. Lu, F., Milios, E.: Robot Pose Estimation in Unknown Environments by Matching 2D Range Scans. Journal of Intelligent and Robotic Systems 18(3), 249–275 (1997)
19. Merke, A., Welker, S., Riedmiller, M.: Line Based Robot Localization under Natural Light Conditions. In: ECAI 2004 Workshop on Agents in Dynamic and Real Time Environments (2004)
20. Moravec, H.P.: Sensor Fusion in Certainty Grids for Mobile Robots. AI Magazine 9(2), 61 (1988)
21. Olson, E., Leonard, J., Teller, S.: Fast iterative alignment of pose graphs with poor initial estimates. In: 2011 IEEE International Conference on Robotics and Automation (ICRA), pp. 2262–2269. IEEE (2006)
22. Riedmiller, M., Braun, H.: A direct adaptive method for faster backpropagation learning: the RPROP algorithm. In: IEEE International Conference on Neural Networks, vol. 1, pp. 586–591 (1993)
23. Thrun, S.: A Probabilistic On-Line Mapping Algorithm for Teams of Mobile Robots. The International Journal of Robotics Research 20(5), 335–363 (2001)
24. Thrun, S.: Probabilistic Algorithms in Robotics. AI Magazine 21(4), 93 (2000)
25. Thrun, S., Liu, Y., Koller, D., Ng, A.Y., Durrant-Whyte, H.: Simultaneous Localization and Mapping with Sparse Extended Information Filters. The International Journal of Robotics Research 23(7-8), 693–716 (2004)
26. Zhang, Z.: Iterative point matching for registration of free-form curves and surfaces. Int. J. Comput. Vision 13(2), 119–152 (1994)