

An Optimal Pose Estimator for Map-based Mobile Robot Dynamic Localization: Experimental Comparison with the EKF

G. A. Borges * M.-J. Aldon T. Gil
Robotics Department
LIRMM, UMR CNRS/Université Montpellier II, n^o. C55060
161, rue ADA. 34392 - Montpellier - Cedex 5 - France
e-mail: {borges, aldon, gil}@lirmm.fr

Abstract

Theoretical solutions based on the matching of 2-D range measurements with a map of the environment have been proposed to solve the robot localization problem. However, most of them have not been experimented in real conditions: the robot was stopped or it moved slowly during range data acquisition, and the environment was supposed to be static. In this paper, we propose and evaluate a dynamic localization method based on feature matching. Experiments carried out in real cluttered indoor environments including people and unknown obstacles show the good performance of the proposed algorithm against the classical solution based on Kalman filtering.

1 Introduction

Estimating the 2-D pose of a vehicle in real environments requires the computation of the geometric configuration of a frame attached to the vehicle with respect to either a local or a global coordinate frame. Relative localization makes generally use of proprioceptive sensors (e.g. odometers, inertial sensors and gyrometers) to estimate the robot motion, and to update the robot configuration with respect to a start position. Most of these sensors provide high-rate measurements, allowing a continuous position estimation. However, dead-reckoning induces errors which increase without bounds. Measurement of known landmarks by exteroceptive sensors (e.g. laser rangefinders, goniometers and video cameras) can be used when an absolute pose update is needed. These measurements concern either artificial beacons, or natural environment features.

Map-based localization uses natural environment features as landmarks. A laser rangefinder is often employed to build a local map which is matched to a global reference map in order to estimate the actual position and heading of

the robot. This approach has been extensively studied for navigation in indoor environments [1]. However, most of the proposed solutions do not take into account some real problems encountered in practical applications:

- They are generally inappropriate to accurate dynamic localization, i.e. localization while the robot is moving, because the range data processing does not consider the measurement error induced by robot motion;
- They make use of the extended Kalman filter (*EKF*) for pose estimation [2][3][4][5]. However, by linearizing a non-linear measurement model around the current predicted pose, the *EKF* may be very sensitive to non-linear aspects of the measurement model and may give suboptimal estimates. Furthermore, as mentioned in [6], this method often uses fixed values of the measurement noise covariance matrices, and most of measurement errors are assumed to have Gaussian distribution. In many practical applications, such assumptions are not true, specially in dynamic environments. Then, the error covariance matrices have a strong influence on the filter convergence.

In a previous paper, we have presented a weighted least-squares-based method to solve the relative motion estimation problem [7]. The matched features between two consecutive range images acquired by a laser rangefinder were used to estimate the robot motion. The method was employed in an iterative manner in order to give better motion estimates by updating the weights at each iteration. Its non-iterative absolute pose estimation version is presented in this paper. In doing so, the matched features are obtained from a local map built from the last range image and a global environment map. It uses a unified representation for heterogeneous features, *i.e.* lines and points. The problem is formulated so as to minimize a weighted least-squares criterion which incorporates feature correspondence measures and position estimates. Thus, the final pose estimates are optimal in the least-squares sense.

*Supported by CAPES - Brasília, Brazil, under grant BEX2280/97-3

In this paper, we are dealing with dynamic localization. Local map building is improved with the incorporation of a direct correction procedure for range images that are deformed during acquisition due to the robot motion. In an experimental comparison carried out in a real cluttered indoor environment including people and unknown, the proposed approach gave better results than the *EKF* solution.

This paper is organized as follows. The pose estimation problem from local and global maps is stated in Section 2. The local map building procedure is described in Section 3. Map matching is described in Section 4. The pose estimation procedures are presented in Section 5, and Section 6 reports the experimental results.

2 Problem statement

We consider a robot equipped with a laser rangefinder navigating on a flat ground in a structured indoor environment. As the robot moves, sensor observations become available at discrete time t , and a local environment map $\mathcal{M}^{\mathcal{L}}(t)$ is extracted. In this work, $\mathcal{M}^{\mathcal{L}}(t)$ is composed of $N_{\mathcal{F}}^{\mathcal{L}}$ geometric features $\mathcal{F}_j^{\mathcal{L}}$ that are linear and ellipsoidal clusters. Its general form is

$$\mathcal{M}^{\mathcal{L}}(t) = \{\mathcal{F}_j^{\mathcal{L}} = (\mathbf{f}_j^{\mathcal{L}}, \mathbf{C}_{\mathbf{f}_j}^{\mathcal{L}}, q_j^{\mathcal{L}}), j = 1, \dots, N_{\mathcal{F}}^{\mathcal{L}}\},$$

where $\mathbf{f}_j^{\mathcal{L}}$ is the j^{th} feature parameter and $\mathbf{C}_{\mathbf{f}_j}^{\mathcal{L}}$ is its covariance matrix. $q_j^{\mathcal{L}}$ is a label that indicates the feature type.

The robot absolute pose at time t is given by $\mathbf{p}(t) = (x(t), y(t), \theta(t))^T$, where $(x(t), y(t))$ is the vehicle 2-D position in the global frame and $\theta(t)$ is its heading. The environment is represented by a global map

$$\mathcal{M}^{\mathcal{G}}(t) = \{\mathcal{F}_i^{\mathcal{G}} = (\mathbf{f}_i^{\mathcal{G}}, \mathbf{C}_{\mathbf{f}_i}^{\mathcal{G}}, q_i^{\mathcal{G}}), i = 1, \dots, N_{\mathcal{F}}^{\mathcal{G}}\}$$

This model may be time varying if a map updating procedure is implanted. The two maps are related by

$$\begin{pmatrix} x^{\mathcal{L}} \\ y^{\mathcal{L}} \end{pmatrix} = \mathbf{R}(\omega) \left\{ \begin{pmatrix} x^{\mathcal{G}} \\ y^{\mathcal{G}} \end{pmatrix} - \mathbf{t} \right\}, \quad (1)$$

with $\mathbf{t} = (x(t), y(t))^T$ and $\omega = \theta(t)$ being respectively the translation and rotation parameters that relate the coordinates $(x^{\mathcal{G}}, y^{\mathcal{G}})^T$ of an absolute point in the global map and its coordinates $(x^{\mathcal{L}}, y^{\mathcal{L}})^T$ in the local map. The rotation matrix $\mathbf{R}(\omega)$ is given by

$$\mathbf{R}(\omega) = \begin{pmatrix} \cos \omega & \sin \omega \\ -\sin \omega & \cos \omega \end{pmatrix}. \quad (2)$$

The problem stated here is: At some time $t^+ > t$, given the local and global environment maps $\mathcal{M}^{\mathcal{L}}(t)$ and $\mathcal{M}^{\mathcal{G}}(t)$, compute an estimate $\hat{\mathbf{p}}(t^+)$ of the robot absolute

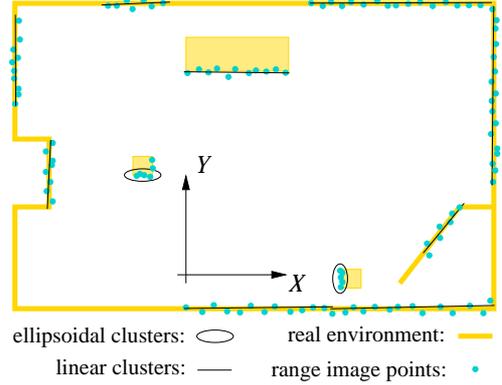


Figure 1: Linear and ellipsoidal geometric features.

pose $\mathbf{p}(t)$ and an associated covariance matrix $C_{\hat{\mathbf{p}}}(t^+)$. The most used approach to solve this problem embodies three phases: pose prediction, map matching and pose estimation. Absolute pose prediction is provided by a dead-reckoning method, like odometry. Features matching from maps $\mathcal{M}^{\mathcal{L}}(t)$ and $\mathcal{M}^{\mathcal{G}}(t)$ is achieved by applying a local procedure where the predicted pose is taken into account. The pose estimate $\hat{\mathbf{p}}(t)$ is obtained from the matched features. In this work, t^+ corresponds to some time after t when a new absolute pose estimate is available after all processing described above and which started at time t .

3 Local map building

In this work we build local maps only from 2-D range images acquired by a rotating laser rangefinder. These maps are composed of geometrical features that are linear and ellipsoidal clusters (see Fig. 1). In real cluttered environments, linear clusters represent vertical planes (e.g. walls and large planar obstacles) when scanned by the laser rangefinder. However, these same environments may contain small objects like furniture that are modeled by linear clusters with short length in the range images. For a long time, short lines have been ignored for localization purposes as their orientation is not reliable. Therefore, the center coordinates given by the mean of the range points that compose the short lines are more stable and may be used as feature description. In this work, short lines are referred to as ellipsoidal clusters.

As we are dealing with dynamic localization, range images are always deformed given that they are acquired during the robot displacement. Thus, we propose a direct range image correction procedure in order to compensate for vehicle motion. Range image correction and feature extraction are applied for local map building, as follows.

3.1 Range image correction

Range images acquired by laser rangefinders have a polar representation $I_P = \{(r_n, \phi_n), n = 1, \dots, N\}$. Another equivalent representation is the cartesian one: $I_C = \{(x_n^S, y_n^S), n = 1, \dots, N\}$, with $x_n^S = r_n \cos \phi_n$ and $y_n^S = r_n \sin \phi_n$. The reference frame of a range image has its origin on the rotation axis of the sensor mirror. The acquisition starts at the angle ϕ_1 and stops at ϕ_N . Each point n represents the coordinates of the intersection point between the laser ray with angular direction ϕ_n and the first obstacle on this direction. The distance of the obstacle at the direction ϕ_n to the sensor reference frame origin is $r_n \geq 0$. The acquisition of a complete range image gets a time interval T_S that is not zero. It means that during a scan, if the robot moves, the range image is deformed. Therefore, it is necessary to perform range image correction before feature extraction.

The proposed correction procedure takes into account the sensor displacement during the acquisition process. Let $\mathbf{p}_n^S = (x_n, y_n, \theta_n)^T$ be the sensor absolute position when the n^{th} point is acquired. In this way, \mathbf{p}_N^S is the sensor pose at the last point, or when the sensor signals end-of-acquisition. At the end-of-acquisition, the local coordinate frame has been displaced of $\mathbf{p}_N^S - \mathbf{p}_n^S$ since the n^{th} point acquisition. However, (x_n^S, y_n^S) represents the coordinates of the n^{th} point when the sensor was at \mathbf{p}_n^S . If we want to recover the coordinates $(\tilde{x}_n^S, \tilde{y}_n^S)$ of the same point when the sensor is on \mathbf{p}_N^S , the sensor displacement is taken into account as

$$\begin{pmatrix} \tilde{x}_n^S \\ \tilde{y}_n^S \end{pmatrix} = \mathbf{R}(\theta_N - \theta_n) \left\{ \begin{pmatrix} x_n^S \\ y_n^S \end{pmatrix} - \begin{pmatrix} x_N - x_n \\ y_N - y_n \end{pmatrix} \right\}. \quad (3)$$

Eq. (3) indicates that it is not necessary to know the sensor absolute pose at each n^{th} point, but its relative displacement $\Delta \mathbf{p}_n^S = \mathbf{p}_N^S - \mathbf{p}_n^S$ to the N^{th} point. Thus, relative pose estimation as that provided by odometry may be used as a guess to $\Delta \mathbf{p}_n^S$. It must be pointed out that $\Delta \mathbf{p}_n^S$ is not the robot displacement, but the sensor displacement.

We know that a well calibrated odometry system is reliable for short time periods. In our implementation, the sensor displacement $\Delta \mathbf{p}_n^S$ is interpolated by a linear relation between $\Delta \mathbf{p}_1^S$ and $\Delta \mathbf{p}_N^S = \mathbf{0}$. Therefore more sophisticated approximations may be developed.

3.2 Feature representation

The main challenge in this work was on choosing a unified feature representation for linear and ellipsoidal features. With such representation, it is possible to solve the pose estimation by means of an embedded least-squares. It means that a least-squares problem which has no solution for separate features may become solvable by achieving simultaneous feature fusion.

We have chosen a unified feature representation that has the following characteristic: linear and ellipsoidal clusters have homogeneous parameters defined in the same Euclidean space. For the case of ellipsoidal clusters we have $\mathbf{f} = (x, y)^T$ which are the coordinates of the cluster center in the Euclidean space. Linear cluster parameters are associated with the line that best fits the cluster points in the least-squares sense. This line has polar parameters that are (ρ, α) . Linear clusters are represented by $\mathbf{f} = (x, y)^T$, with $x = \rho \cos \alpha$ and $y = \rho \sin \alpha$ being the coordinates of the point on the line that is closest to the Cartesian coordinates origin. In order to identify each feature type, a label q is defined. Associated to each feature, we have a covariance matrix \mathbf{C}_f that is estimated from the data fit residuals by first-order covariance propagation. Therefore, we have a compact feature representation that is $\mathcal{F} = \{\mathbf{f}, \mathbf{C}_f, q\}$.

We can show that given the relation between the local and global reference frames of Eq. (1), the local (\mathbf{f}^L) and global (\mathbf{f}^G) feature parameters are related by

$$\mathbf{f}^L = \mathbf{R}(\omega) \cdot (\mathbf{f}^G - \mathbf{W}^G \mathbf{t}), \quad (4)$$

with

$$\mathbf{W}^G = \begin{pmatrix} a^G & c^G \\ c^G & b^G \end{pmatrix}. \quad (5)$$

The parameters a^G , b^G and c^G are recovered from the compact parameter representation $\mathcal{F}^G = \{\mathbf{f}^G, \mathbf{C}_f^G, q^G\}$. If q^G denotes an ellipsoidal cluster: $a^G = b^G = 1$ and $c^G = 0$. If q^G denotes a linear cluster: $a^G = \cos \alpha^G$, $b^G = \sin \alpha^G$ and $c^G = a^G b^G$, with $\alpha^G = \arctan(y^G/x^G)$.

3.3 Feature extraction

Feature extraction from the rectified range image is achieved by applying a clustering algorithm to find only linear clusters [8]. This algorithm embodies a classical fuzzy clustering algorithm in a split-and-merge framework. It avoids the need to have an initial guess for the number of clusters. During the split phase it recursively cuts the entire range image into two linear clusters at each iteration, and stops when all actual clusters satisfy a validity criterion. In the merge phase clusters are fused if the new cluster also satisfies the validity criterion. Further details are presented in [8]. This algorithm gives much better results than the classical iterative end-point fit, usually employed for line extraction in video images. Some of the extracted linear clusters will become ellipsoidal clusters given the line length. In experiments, we have chosen as ellipsoidal clusters all linear clusters whose length was less than 0.5 m.

4 Map matching

Map matching is essential for pose estimation. In this phase we attempt to find a list of local and global features

that correspond. The usual approach to find feature correspondences is to perform a local matching by using the Mahalanobis distance. In order to verify if two features $\mathcal{F}_j^{\mathcal{L}} = (\mathbf{f}_j^{\mathcal{L}}, \mathbf{C}_{\mathbf{f}_j^{\mathcal{L}}}, q_j^{\mathcal{L}}) \in \mathcal{M}^{\mathcal{L}}(t)$ and $\mathcal{F}_i^{\mathcal{G}} = (\mathbf{f}_i^{\mathcal{G}}, \mathbf{C}_{\mathbf{f}_i^{\mathcal{G}}}, q_i^{\mathcal{G}}) \in \mathcal{M}^{\mathcal{G}}(t)$ match, this statistic measure takes into account the features parameters and covariances, an initial guess of the absolute pose $\hat{\mathbf{p}}(t)$, and its associated covariance $\mathbf{C}_{\hat{\mathbf{p}}}(t)$. As in classical approaches, $\hat{\mathbf{p}}(t)$ and $\mathbf{C}_{\hat{\mathbf{p}}}(t)$ are given by odometry. Thus, for each feature pair of the same type (e.g. $q_i^{\mathcal{G}} = q_j^{\mathcal{L}}$), a match is verified if

$$\mathbf{r}_{ij}^T(t) \mathbf{C}_{r_{ij}}^{-1}(t) \mathbf{r}_{ij}(t) \leq Q, \quad (6)$$

with $\mathbf{r}_{ij}(t) = \mathbf{f}_j^{\mathcal{L}} - \mathbf{R}(\hat{\omega}(t)) \cdot (\mathbf{f}_i^{\mathcal{G}} - \mathbf{W}^{\mathcal{G}} \hat{\mathbf{t}}(t))$ being the feature prediction residual and

$$\begin{aligned} \mathbf{C}_{r_{ij}} = & \left(\frac{\partial \mathbf{r}_{ij}(t)}{\partial \hat{\mathbf{p}}(t)} \right) \mathbf{C}_{\hat{\mathbf{p}}}(t) \left(\frac{\partial \mathbf{r}_{ij}(t)}{\partial \hat{\mathbf{p}}(t)} \right)^T + \\ & \left(\frac{\partial \mathbf{r}_{ij}}{\partial \mathbf{f}_i^{\mathcal{G}}} \right) \mathbf{C}_{\mathbf{f}_i^{\mathcal{G}}} \left(\frac{\partial \mathbf{r}_{ij}}{\partial \mathbf{f}_i^{\mathcal{G}}} \right)^T + \left(\frac{\partial \mathbf{r}_{ij}}{\partial \mathbf{f}_j^{\mathcal{L}}} \right) \mathbf{C}_{\mathbf{f}_j^{\mathcal{L}}} \left(\frac{\partial \mathbf{r}_{ij}}{\partial \mathbf{f}_j^{\mathcal{L}}} \right)^T \end{aligned}$$

being its covariance estimated by first-order covariance propagation. Q is a validation gate that is determined assuming that $\mathbf{r}_{ij}^T(t) \mathbf{C}_{r_{ij}}^{-1}(t) \mathbf{r}_{ij}(t)$ follows a χ_2^2 distribution. Usually $Q = 4.61$, that corresponds to a 90 % confidence region [9].

All N_{Ω} matched features are stored in a list $\Omega(t)$ where the k^{th} entry is $\Omega_k = (\mathcal{F}_i^{\mathcal{G}}, \mathcal{F}_j^{\mathcal{L}}, \mu_k)$ such that Eq. (6) is verified. $\mu_k \geq 0$ is a correspondence measure, that is determined in function of the pose estimation approach.

5 Pose estimation

Pose estimation is accomplished from the list $\Omega(t)$ of matched features. The extended Kalman filter (EKF), that is a powerful tool for state estimation of non-linear systems [9], is the most used pose estimator. However, it gives suboptimal estimates and its convergence depends of consistent estimates of the noise covariance matrices. To overcome these problems, we present a weighted least-squares (WLS) approach, introduced in [7], that gives optimal pose estimate in the least-squares sense.

The main advantage of using the unique feature representation described in Section 3.2 is that a unique cost function can embody the different features for optimal estimation. This is what we have done in a WLS framework. Hence, we minimize the following cost function:

$$J(\Omega, \hat{\mathbf{t}}, \hat{\omega}) = \sum_{k=1}^{N_{\Omega}} \mu_k \mathbf{r}_k^T \mathbf{r}_k. \quad (7)$$

where $\mathbf{r}_k = \mathbf{f}_j^{\mathcal{L}} - \mathbf{R}(\hat{\omega}) \cdot (\mathbf{f}_i^{\mathcal{G}} - \mathbf{W}^{\mathcal{G}} \hat{\mathbf{t}})$ is the residual of the k^{th} matched features of $\Omega(t)$. Being \mathbf{r}_k a non-linear

function of $\hat{\mathbf{t}}$ and $\hat{\omega}$, it is possible that Eq. (7) has more than one local minimum. The first step for the minimization of J is to find all local minima. In doing so, we solve

$$\frac{\partial J}{\partial \hat{\mathbf{t}}} = \mathbf{0} \text{ and } \frac{\partial J}{\partial \hat{\omega}} = 0. \quad (8)$$

This results in the following non-linear system:

$$\mathbf{M} \hat{\mathbf{t}} - \mathbf{N} \hat{\beta} - \mathbf{q} = \mathbf{0}, \quad (9)$$

$$\hat{\beta}^T \mathbf{r} + \hat{\beta}^T \mathbf{P} \hat{\mathbf{t}} = 0, \quad (10)$$

with $\hat{\beta}^T = (\cos \hat{\omega} \quad \sin \hat{\omega})$. \mathbf{M} , \mathbf{N} and \mathbf{P} are 2×2 matrices, and \mathbf{q} and \mathbf{s} are 2×1 vectors. From equations (9) and (10) we obtain a trigonometric equation on $\hat{\omega}$ which has four solutions $\hat{\omega}_m$, $m = 1, \dots, 4$. The corresponding solutions $\hat{\mathbf{t}}_m$ are obtained from Eq. (9). Therefore, each $(\hat{\mathbf{t}}_m, \hat{\omega}_m)$ pair is a local minimum of J . The optimal WLS pose estimate $\hat{\mathbf{p}}_{WLS}^T(t^+)$ is found by evaluating J for each local minimum, and choosing the one corresponding to the lowest cost function value. Further details on the development and solution of Eqs. (9) and (10) are given in [7]. This is a non-iterative direct solution to the WLS criterion.

As for the EKF, pose estimate uncertainty is given by a covariance matrix $\mathbf{C}_{\hat{\mathbf{p}}_{WLS}}(t^+)$. This matrix is estimated by applying the covariance propagation methodology described in [10]. From the mathematical development reported in [10], we have

$$\mathbf{C}_{\hat{\mathbf{p}}_{WLS}}(t^+) = \left(\frac{\partial \mathbf{g}}{\partial \hat{\mathbf{p}}_{WLS}} \right)^{-1} \cdot \Lambda \cdot \left(\frac{\partial \mathbf{g}}{\partial \hat{\mathbf{p}}_{WLS}} \right)^{-1}, \quad (11)$$

with $\mathbf{g} = \partial J(\Omega, \hat{\mathbf{p}}_{WLS}) / \partial \hat{\mathbf{p}}_{WLS}(t^+)$, and

$$\begin{aligned} \Lambda = & \sum_{k=1}^{N_{\Omega}} \frac{\partial \mathbf{g}}{\partial \mathbf{r}_k(t^+)} \cdot \mathbf{C}_{\mathbf{r}}(t^+) \cdot \frac{\partial \mathbf{g}}{\partial \mathbf{r}_k(t^+)}^T + \\ & \sum_{i \in \Omega(t)} \frac{\partial \mathbf{g}}{\partial \mathbf{f}_i^{\mathcal{L}}} \cdot \mathbf{C}_{\mathbf{f}_i^{\mathcal{L}}} \cdot \frac{\partial \mathbf{g}}{\partial \mathbf{f}_i^{\mathcal{L}}}^T + \sum_{j \in \Omega(t)} \frac{\partial \mathbf{g}}{\partial \mathbf{f}_j^{\mathcal{G}}} \cdot \mathbf{C}_{\mathbf{f}_j^{\mathcal{G}}} \cdot \frac{\partial \mathbf{g}}{\partial \mathbf{f}_j^{\mathcal{G}}}^T. \end{aligned}$$

In the above equations, $\mathbf{r}_k(t^+)$ is the residual of the k^{th} matched features after estimation, and $\mathbf{C}_{\mathbf{r}}(t^+)$ is its associated covariance matrix estimated as $\mathbf{C}_{\mathbf{r}}(t^+) = \sum_{k=1}^{N_{\Omega}} \mu_k (\mathbf{r}_k \cdot \mathbf{r}_k^T) / \sum_{k=1}^{N_{\Omega}} \mu_k$. $\mathbf{f}_i^{\mathcal{L}}$ and $\mathbf{f}_j^{\mathcal{G}}$ are the local and global feature vectors of the k^{th} correspondence.

The final pose estimate $\hat{\mathbf{p}}(t^+)$ is then computed by fusing the WLS and the odometric estimates as

$$\begin{aligned} \hat{\mathbf{p}}(t^+) = & \mathbf{C}_{\hat{\mathbf{p}}}(t) (\mathbf{C}_{\hat{\mathbf{p}}_{WLS}}(t^+) + \mathbf{C}_{\hat{\mathbf{p}}}(t))^{-1} \hat{\mathbf{p}}_{WLS}(t^+) \\ & + \mathbf{C}_{\hat{\mathbf{p}}_{WLS}}(t^+) (\mathbf{C}_{\hat{\mathbf{p}}_{WLS}}(t^+) + \mathbf{C}_{\hat{\mathbf{p}}}(t))^{-1} \hat{\mathbf{p}}(t), \end{aligned} \quad (12)$$

and the associated covariance matrix is

$$\mathbf{C}_{\hat{\mathbf{p}}}(t^+) = \mathbf{C}_{\hat{\mathbf{p}}_{WLS}}(t^+) (\mathbf{C}_{\hat{\mathbf{p}}_{WLS}}(t^+) + \mathbf{C}_{\hat{\mathbf{p}}}(t))^{-1} \mathbf{C}_{\hat{\mathbf{p}}}(t). \quad (13)$$

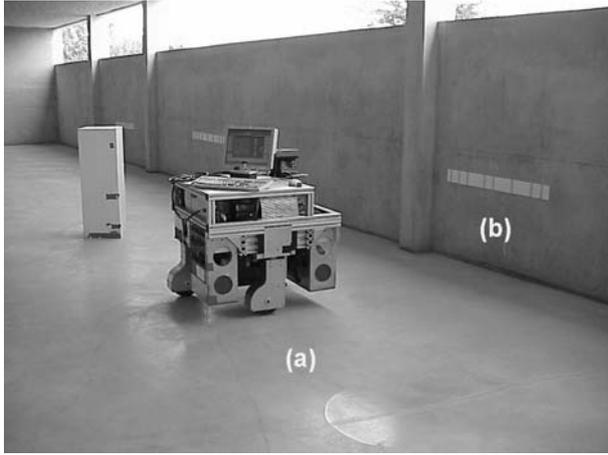


Figure 2: (a) Omni robot and (b) artificial beacon.

6 Experimental validation

Experimental validation was carried out in a real environment with the omnidirectional robot Omni [11]. In this environment, we have installed artificial identifiable beacons in strategic placements. A reference localization system that uses a laser goniometer for beacon detection and identification [12] was implanted on the robot. When a beacon is detected, the relative robot pose with respect to the beacon is estimated by triangulation. With the identification code, the beacon absolute pose is recovered from a lookup table. Thus, a reference robot absolute pose $\mathbf{p}^*(t)$ is estimated and used only to evaluate the performance of the localization algorithms. The errors on the estimate $\mathbf{p}^*(t)$ are less than 0.5° for heading, and 2 cm for position. Figure 2 shows a picture of the robot Omni (a) in front of an artificial beacon (b).

The rangefinder used for local map building is an *2D30* model manufactured by *IBEO Lasertechnik*. Its maximum range is 30 m , with a precision of $\pm 5\text{ cm}$. This sensor provides scans in a 270° angular field with a resolution of 0.6° . The range image rate is of 8 images per second.

We have evaluated two absolute localization algorithms that differ only by the estimation phase. One of them uses the *EKF* and the other delivers a *WLS* pose estimate. Their pose estimates are compared with those provided by the beacon-based localization system. For the *WLS* algorithm, the weight μ_k for each matched feature was computed as $\mu_k = \exp^{h_k \cdot \log(0.5)/Q}$, where h_k is the Mahalanobis distance between the matched features and $Q = 4.61$ (Eq. (6)). It means that all weights are in the interval $[0.5, 1]$. More reliable matches have weights close to 1.

The robot pose is updated by odometry between two estimates provided by the localization algorithms. The local map is built each 8 range images, resulting in an absolute

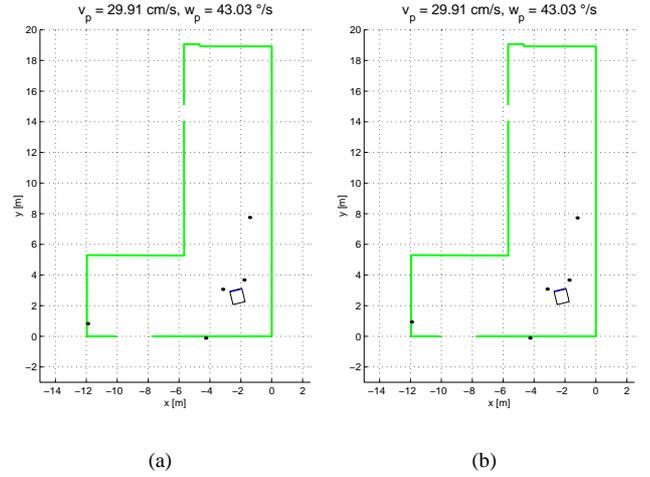


Figure 3: The points of a range image acquired during motion: (a) before and (b) after motion rectification. The global map is provided for comparison.

Table 1: EKF and WLS performance measures

	x	y	θ
$\bar{\varepsilon}$ (EKF)	1.49 cm	-2.91 cm	-1.47°
σ_ε (EKF)	6.08 cm	6.82 cm	6.12°
$\bar{\varepsilon}$ (WLS)	0.43 cm	-2.30 cm	-0.49°
σ_ε (WLS)	5.81 cm	6.37 cm	0.87°

pose estimate each second. As the main purpose of this experiment is not on the validation of a map building technique, we used a global map that was built off line.

The experiment consisted in guiding the robot through a complex trajectory by the mean of a joystick. Some people participated in the experiment as mobile obstacles. The total experimentation time was of one minute and 61 pose estimates were provided by each localization algorithm. As a validation experiment for dynamic localization, the robot was not stopped for local map acquisition. Figure 3(a) shows a range image acquired during a motion executed with a translational speed of 29.91 cm/s and a rotational speed of $43.03^\circ/s$. The global map is provided for comparison. As we can see, this image has an important deformation with respect to the global map. After applying the correction procedure, the rectified image seems to best fit the environment model (Fig. 3(b)).

In order to compare the EKF and WLS algorithms, we have defined two performance measures based on the pose estimates offset with respect to the beacon-based reference poses $\mathbf{p}^*(t)$. The offset is defined as $\varepsilon(t^+) = \mathbf{p}^*(t^+) - \hat{\mathbf{p}}(t^+)$. As performance measures we have the mean offset, denoted as $\bar{\varepsilon}$ and used as bias indicator, and the offset stan-

standard deviation σ_ε . These measures are taken for each pose component x , y and θ , and presented in Table 1 for both algorithms. It can be verified that the WLS estimates are less biased than the EKF ones. The position offset corresponding to x and y presented almost the same variance. However, the heading estimates provided by WLS are much better than the ones given by the EKF.

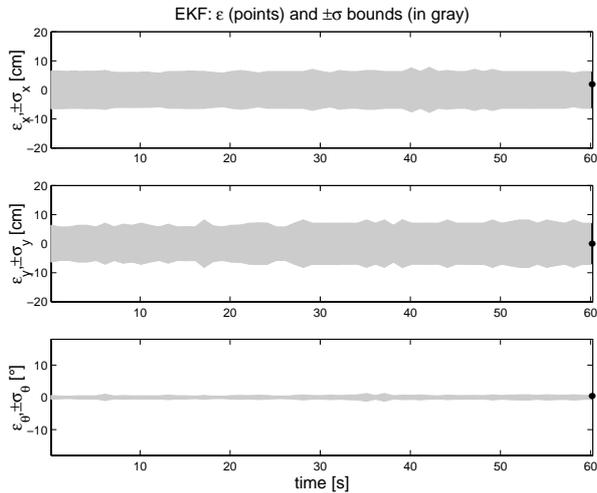


Figure 4: Extended Kalman filter pose offset results.

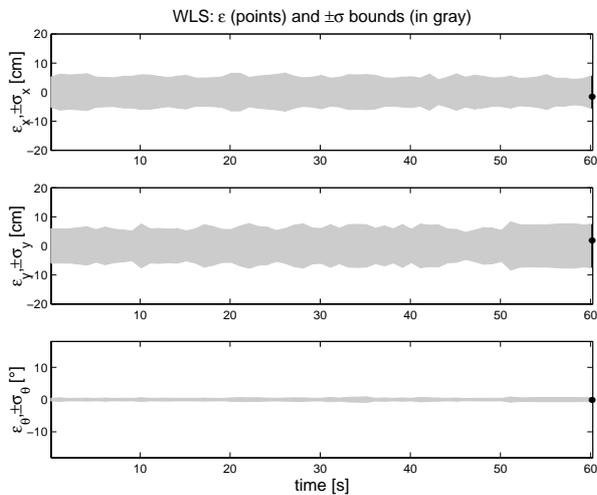


Figure 5: Weighted least squares pose offset results.

The plots of Figures 4 and 5 show respectively the instantaneous values of $\varepsilon(t^+)$ for this experiment. The associated standard deviation bounds $\pm\sigma$ extracted from estimated covariance matrices are also plotted as gray intervals. The $\pm\sigma$ bounds for the position seem to be well estimated by both algorithms. However, the same was not true for heading. Only the WLS algorithm gave more consistent results, with most of the estimation offsets between $\pm\sigma$.

7 Conclusion

This paper presented an absolute pose estimation algorithm for dynamic mobile robot localization. The absolute pose estimate is optimal in the weighted least-squares sense. This method does not suffer from the non-convergence problems associated to linearization as the classical *EKF* approach. We proposed a direct correction procedure to compensate for range image deformation due to robot motion. Experimental results demonstrated the good performance of this solution. We believe that the non-convergence of the EKF is due to the well known problems related to model linearization.

As future work, we plan to evaluate a more robust version by converting the weighted least-squares into an iterative M-estimator. In doing so, the weights are updated following robust scale estimates.

References

- [1] J. Borenstein, H. R. Everett, and L. Feng. *Navigating Mobile Robots: Systems and Techniques*. A K Peters, Wellesley, Massachusetts, 1996.
- [2] Tobias Einsele. Real-time self-localization in unknown indoor environments using a panorama laser range finder. In *Proc. of IEEE/RSJ IROS*, pages 697–702, 1997.
- [3] John J. Leonard, Hugh F. Durrant-Whyte, and Ingemar J. Cox. Dynamic map building for an autonomous mobile robot. *The International Journal of Robotics Research*, 11(4):286–298, August 1992.
- [4] Artur Arsenio and M. Isabel Ribeiro. Active range sensing for mobile robot localization. In *Proc. of IEEE/RSJ IROS*, October 1998.
- [5] H. Yang, H. Chung, K. C. Park, and J. G. Lee. A rotating sonar and a differential encoder data fusion for map-based dynamic localization. In *Proc. of IFAC/ Intelligent Autonomous Vehicles*, March 1998.
- [6] L. Jetto and S. Longhi. Development and experimental validation of an adaptive kalman filter for the localization of mobile robots. *IEEE Transactions on Robotics and Automation*, 15(2):219–229, 1999.
- [7] G. A. Borges and M.-J. Aldon. Motion estimation by iterative 2-d features matching in range images. In *IEEE International Conference on Robotics and Automation*, April 2000.
- [8] G. A. Borges and M.-J. Aldon. A split-and-merge segmentation algorithm for line extractions in 2-d range images. In *15th International Conference on Pattern Recognition*, September 2000.
- [9] Y. Bar-Shalom and X.-R. Li. *Estimation and Tracking: Principles, Techniques and Software*. Norwood: Artech House, 1987.
- [10] R.M. Haralick. Propagating covariances in computer vision. In *International Conference on Pattern Recognition*, pages 493–498, 1994.
- [11] Yann Le Corre. *Conception et Commande d'un Robot Omnidirectionnel*. PhD thesis, Université Montpellier II, LIRMM, 161 rue ADA, 34392, Montpellier, Cedex 5, France, Décembre 1998.
- [12] M. J. Aldon and M. Benoit. Localization system of the hospital trans robot first. In *Proc. of IFAC Intelligent Autonomous Vehicles*, pages 244–249, June 1995.