Motion Estimation by Iterative 2-D Features Matching in Range Images

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

Abstract

This article describes an iterative algorithm for relative motion estimation from 2-D range images. The matching is achieved in the geometric feature domain represented by straight lines and ellipsoidal clusters. Using infinite length straight lines instead of line segments as features, the algorithm attempts to achieve robustness to partial occlusions. The motion estimates and the feature correspondence measures are determined in order to minimize a cost function. The performance of this algorithm was evaluated with experiments carried out in real cluttered indoor environments.

1 Introduction

During the last decade, map-based positioning techniques using laser range sensors have been extensively studied to solve the problem of autonomous mobile robot navigation in indoor environments. Absolute localization methods are generally based on the matching of a current local map provided by the sensor with a stored 2-D global reference map. When a correct match is found, the actual position and orientation of the robot are computed. A more complete review on map-based absolute localization approaches for mobile robots is presented in [1]. The same pose estimation methods may be applied to the relative motion estimation problem. In this paper, we are interested in matching two consecutive range images by using geometric features.

The general approach used for position estimation by geometric features matching is the Extended Kalman Filter (*EKF*). The *EKF* is used to track features in consecutive range images, resulting in a local matching procedure. Using linearization 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, measurement errors are assumed to have Gaussian distribution and the error covariance matrices have a strong influence on the filter convergence. Alternative approaches for position estimation by features matching in range scan images use dynamic programming [2] and clustering techniques [3].

Matching techniques of range images can be classified according to the kind of feature used to solve the correspondence problem: measured points [4], line segments [2], and geometric beacons like corners [5, 6]. As indoor environments are rich in polygonal objects, line segments characterized by end-points are often used as basic features. However, as partial occlusions frequently occur in real cluttered environments, the length of the line segments may change considerably, or new line segments can appear. Thus, if no match is found the problem becomes ill-conditioned. We believe that features like infinite length straight lines are less sensitive to partial occlusion. These features have a polar representation given by their inclination and the distance to the local frame origin. However, the polar parameters of short length straight lines, as those generated by small polyhedral obstacles, are very sensitive to the robot's pose. We decided to label these short lines as ellipsoidal features, described by the coordinates of their center of gravity.

The method presented in this paper matches two consecutive range images using the above fundamental features in an iterative manner. As in *EKF*-based approaches, the proposed algorithm starts from an initial motion estimate and its associated uncertainty given by dead-reckoning. The problem is formulated so as to minimize a cost function incorporating feature correspondence measures and motion estimates. Using the alternative optimization approach, the feature correspondence measures and the motion estimates are

*Work supported by a grant from CAPES - Brasília, Brazil

updated at each iteration. Experiments carried out in real cluttered indoor environments show the good performance of the proposed algorithm.

This paper is organized as follows. Section 2 describes the relative motion estimation problem and the features representation. The motion estimation algorithm is presented in Section 3, and the implementation aspects are discussed in Section 4. Section 5 presents the experimental results and Section 6 contains the conclusions.

2 Problem Statement

We consider that a robot equipped with a laser radar moves on a flat ground in a structured indoor environment. All acquired 2-D range images are formed by a sequence of points in the Cartesian coordinate system (X_L, Y_L) of the sensor. They are represented in the polar form $I = \{\rho_s, \varphi_s\}_{s=1}^{N}$. ρ_s is the distance between the origin of (X_L, Y_L) and the nearest obstacle in the scanning direction given by the angle φ_s with respect to X_L. For convenience, all motion estimates are referred to the (X_L, Y_L) coordinate system.

As the vehicle moves, a sequence of 2-D range images are acquired by the ladar at different absolute poses. The problem stated in this work is to find the vehicle's motion between two consecutive absolute poses P_{k-1} and P_k by matching the corresponding range images I_{k-1} and I_k. Due to the robot's movement, a static point in the absolute system is observed in images I_{k-1} and I_k at the cartesian coordinates (x_{k-1}, y_{k-1}) and (x_k, y_k), respectively. We establish that the motion is given by a translation $\mathbf{t} = (t_x, t_y)^T$ and a rotation angle ω such that

$$\begin{pmatrix} x_k \\ y_k \end{pmatrix} = \mathbf{R}(\omega) \left\{ \begin{pmatrix} x_{k-1} \\ y_{k-1} \end{pmatrix} - \mathbf{t} \right\}, \qquad (1)$$

with

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

The matching of images I_{k-1} and I_k is achieved using linear (straight lines) and ellipsoidal features. As these features are extracted from range images in the form of sets of points with a special organization, they are treated as clusters. A linear cluster is represented by the polar parameters ρ_l and α_l of the line that best fits the set of points (see Fig. 1). Let $\{x_i, y_i\}_{i=1}^{n}$ a set of points in the Cartesian coordinate system. The straight line parameters that best fit these points are given by [4]

$$\rho_l = \bar{x}\cos\alpha_l + \bar{y}\sin\alpha_l, \qquad (3)$$

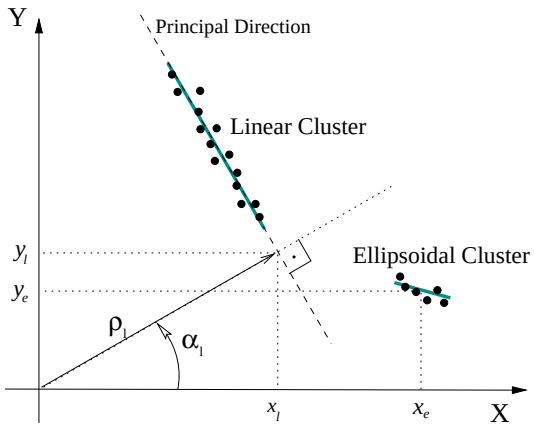

Figure 1: Features description.

$$\alpha_l = \frac{1}{2}\arctan\left(\frac{-2S_{xy}}{S_{yy} - S_{xx}}\right), \qquad (4)$$

with

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i, \ \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i \qquad (5)$$

$$S_{xx} = \sum_{i=1}^{n}(x_i - \bar{x})^2, \ S_{yy} = \sum_{i=1}^{n}(y_i - \bar{y})^2, \qquad (6)$$

$$S_{xy} = \sum_{i=1}^{n}(x_i - \bar{x})\cdot(y_i - \bar{y}). \qquad (7)$$

Ellipsoidal clusters are defined by the components $x_e = \bar{x}$ and $y_e = \bar{y}$ of their center of gravity (Fig. 1).

A common quality measure for all clusters is the empirical standard deviation of the points set with respect to the principal direction line. This parameter is given by

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\rho_l - x_i\cos\alpha_l + y_i\sin\alpha_l)^2}. \qquad (8)$$

3 Motion Estimation

The motion estimation algorithm assumes that initial estimates $\hat{\mathbf{t}}$ and $\hat{\omega}$ of $\mathbf{t}$ and $\boldsymbol{\omega}$ are given by a dead-reckoning method. It is well known that dead-reckoning has errors that grow with time. Therefore, for short periods of time, systematic errors can be well modeled and error bounds could be directly obtained from uncertainty ellipses. Our approach attempts to track features between two consecutive range images I_{k-1} and I_k given initial estimates $\hat{\mathbf{t}}_0$ and $\hat{\omega}_0$ and error bounds $\boldsymbol{\Delta}\mathbf{t}_0$ and $\Delta\omega_0$.

We formulate the motion estimation and the feature tracking as an optimization problem. In this context, we iteratively reduce a cost function which incorporates the feature correspondence measures and evaluates the motion estimates. This type of criterion is extensively used by the pattern recognition community for the development of robust clustering methods. Therefore, we have chosen a cost function derived from the Possibilistic C-Means algorithm (PCM) [7], and given by

$$\begin{aligned} J &= \sum_{i=1}^{N_{k-1}} \sum_{j=1}^{N_k} \mu_{ij} d^2(\mathbf{f}_i, \mathbf{f}_j, \hat{\mathbf{t}}, \hat{\omega}) \\ &\quad + \sum_{i=1}^{N_{k-1}} \eta_i \sum_{j=1}^{N_k} \mu_{ij} \left(\log(\mu_{ij}) - 1 \right), \end{aligned} \tag{9}$$

where μ_{ij} and $d(\mathbf{f}_i, \mathbf{f}_j, \hat{\mathbf{t}}, \hat{\omega})$ are the correspondence and distance measures between the features $\mathbf{f}_i$ and $\mathbf{f}_j$, given the current estimates $\hat{\mathbf{t}}$ and $\hat{\omega}$. In order to minimize the effects due to non-normalization, the representation of each extracted line is changed to $\mathbf{l} = (x_l, y_l)^T$, with $x_l = \rho_l \cos(\alpha_l)$ and $y_l = \rho_l \sin(\alpha_l)$ (see Fig. 1). This allows us to have a uniform cluster representation in the Euclidian space in the form $\mathbf{f}_p = (x_p, y_p)^T$. N_{k-1} and N_k are the number of features in I_{k-1} and I_k, respectively, and η_i is a parameter which is feature dependent.

The distance function $d(\mathbf{f}_i, \mathbf{f}_j, \hat{\mathbf{t}}, \hat{\omega})$ must be zero for two corresponding features $\mathbf{f}_i$ and $\mathbf{f}_j$ related by the real motion $\hat{\mathbf{t}} = \mathbf{t}$ and $\hat{\omega} = \omega$. Two corresponding features are defined as being generated by the same static object in the environment. Therefore, given a feature $\mathbf{f}_i = (x_i, y_i)^T$ in image I_{k-1} and its corresponding $\mathbf{f}_j = (x_j, y_j)^T$ in image I_k, we can show from eq. (1) that their parameters are related by

$$\begin{pmatrix} x_j \\ y_j \end{pmatrix} = \boldsymbol{\lambda}(\mathbf{f}_i, \mathbf{t}, \omega), \tag{10}$$

$$= \mathbf{R}(\omega) \left\{ \begin{pmatrix} x_i \\ y_i \end{pmatrix} - \begin{pmatrix} a_i^2 & c_i \\ c_i & b_i^2 \end{pmatrix} \mathbf{t} \right\}. \tag{11}$$

In this approach, we do not consider correspondence between a linear and an ellipsoidal feature. For two corresponding linear clusters, we have $a_i = \cos\alpha_i$, $b_i = \sin\alpha_i$ and $c_i = a_i b_i$, where α_i is given by eq. (4). For ellipsoidal clusters, $a_i = b_i = 1$ and $c_i = 0$. Therefore, a suitable choice for the distance function between two features is

$$d(\mathbf{f}_i, \mathbf{f}_j, \hat{\mathbf{t}}, \hat{\omega}) = \left\| \mathbf{f}_j - \boldsymbol{\lambda}(\mathbf{f}_i, \hat{\mathbf{t}}, \hat{\omega}) \right\|, \tag{12}$$

where $\|\cdot\|$ is the vector norm.

In the same way as in the PCM algorithm, minimization of J is achieved using the alternative optimization approach. Firstly, we consider the current estimates $\hat{\mathbf{t}}$ and $\hat{\omega}$ as constants, and we determine μ_{ij} such that

$$\frac{\partial J}{\partial \mu_{ij}} = 0. \tag{13}$$

Thus, μ_{ij} is given by a Gaussian function:

$$\mu_{ij} = \exp\left(-\frac{d^2(\mathbf{f}_i, \mathbf{f}_j, \hat{\mathbf{t}}, \hat{\omega})}{\eta_i} \right). \tag{14}$$

It can be pointed out that d is a real-valued function and $0 \leq \mu_{ij} \leq 1$ for $\eta_i > 0$. As shown by eq. (14), the parameter η_i controls the width of the Gaussian function. So, it is important to determine suitable values for η_i. If $\eta_i \ll d^2(\mathbf{f}_i, \mathbf{f}_j, \hat{\mathbf{t}}, \hat{\omega})$, two corresponding straight lines $\mathbf{f}_i$ and $\mathbf{f}_j$ that have a small value for the distance measure $d^2(\mathbf{f}_i, \mathbf{f}_j, \hat{\mathbf{t}}, \hat{\omega})$ will receive a small correspondence measure μ_{ij}. If η_i is great enough such that $\eta_i \gg d^2(\mathbf{f}_i, \mathbf{f}_j, \hat{\mathbf{t}}, \hat{\omega})$ for two straight lines that do not correspond, μ_{ij} will be close to 1.0. This kind of mistake is not suitable and η_i must be carefully chosen. This issue is discussed in Section 4.

The estimates $\hat{\mathbf{t}}$ and $\hat{\omega}$ are determined by considering that μ_{ij} is constant and by solving

$$\frac{\partial J}{\partial \hat{\mathbf{t}}} = \mathbf{0} \text{ and } \frac{\partial J}{\partial \hat{\omega}} = 0. \tag{15}$$

This results in the following non-linear system:

$$\mathbf{M}\hat{\mathbf{t}} - \mathbf{N}\hat{\boldsymbol{\beta}} - \mathbf{q} = \mathbf{0}, \tag{16}$$

$$\hat{\boldsymbol{\beta}}^T \mathbf{r} + \hat{\boldsymbol{\beta}}^T \mathbf{P}\hat{\mathbf{t}} = 0, \tag{17}$$

with $\hat{\boldsymbol{\beta}}^T = \begin{pmatrix} \cos\hat{\omega} & \sin\hat{\omega} \end{pmatrix}$, and

$$\mathbf{M} = \sum_{i=1}^{N_{k-1}} \sum_{j=1}^{N_k} \mu_{ij} \begin{pmatrix} c_i^2 + a_i^4 & c_i(a_i^2 + b_i^2) \\ c_i(a_i^2 + b_i^2) & c_i^2 + b_i^4 \end{pmatrix}, \tag{18}$$

$$\mathbf{N} = \sum_{i=1}^{N_{k-1}} \sum_{j=1}^{N_k} \mu_{ij} \begin{pmatrix} -a_i^2 x_j - c_i y_j & a_i^2 y_j - c_i x_j \\ -c_i x_j - b_i^2 y_j & c_i y_j - b_i^2 x_j \end{pmatrix}, \tag{19}$$

$$\mathbf{q} = \sum_{i=1}^{N_{k-1}} \sum_{j=1}^{N_k} \mu_{ij} \begin{pmatrix} c_i y_i + a_i^2 x_i \\ c_i x_i + b_i^2 y_i \end{pmatrix}, \tag{20}$$

$$\mathbf{r} = \sum_{i=1}^{N_{k-1}} \sum_{j=1}^{N_k} \mu_{ij} \begin{pmatrix} -x_i y_j + y_i x_j \\ -y_i y_j - x_i x_j \end{pmatrix}, \tag{21}$$

$$\mathbf{P} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \mathbf{N}^T. \tag{22}$$

From eqs. (16) and (17) we have

$$\hat{\boldsymbol{\beta}}^T \mathbf{P}\mathbf{M}^{-1}\mathbf{N}\hat{\boldsymbol{\beta}} + \hat{\boldsymbol{\beta}}^T \left(\mathbf{r} + \mathbf{P}\mathbf{M}^{-1} \right) = 0. \tag{23}$$

Eq. (23) may be transformed in a fourth order polynomial in $z = \sin\hat{\omega}$. Using a numerical method for finding polynomial roots, four candidates for $\hat{\omega}$ are determined. We can select $m \leq 4$ solutions z_q, $q = 1, \ldots, m$, such that $-1 \leq z_q \leq 1$ and $z_q \in \mathcal{R}$. As $\sin\hat{\omega} = \sin(\pi - \hat{\omega})$, from the selected z_q we have $2m$ candidates for $\hat{\omega}$. The final choice for $\hat{\omega}$ is the one which minimizes J. $\hat{\mathbf{t}}$ is given by eq. (16).

4 Implementation

The final motion estimation algorithm is written as:

1. Initialization:
   (a) The laser radar provides two range images I_{k-1} and I_k taken at consecutive absolute poses P_{k-1} and P_k;
   (b) Dead-reckoning provides initial estimates $\hat{\mathbf{t}}_0$ and $\hat{\omega}_0$ of the motion between P_{k-1} and P_k, as well as the associated error bounds $\boldsymbol{\Delta}\mathbf{t}_0$ and $\Delta\omega_0$.
   (c) Extract features from images I_{k-1} and I_k;
   (d) Initialize the iteration number $n := 0$.
2. Evaluate the distance functions as given by eq. (12) and the current estimates $\hat{\mathbf{t}}_n$ and $\hat{\omega}_n$.
3. Determine suitable values for η_i.
4. Update the correspondence measures μ_{ij} (eq. 14).
5. $n := n + 1$.
6. From equations (16) and (17), determine the new estimates $\hat{\mathbf{t}}_n$ and $\hat{\omega}_n$.
7. Test for the algorithm stabilization: if $\left\|\hat{\mathbf{t}}_n - \hat{\mathbf{t}}_{n-1}\right\| \leq \epsilon_t$ and $\|\hat{\omega}_n - \hat{\omega}_{n-1}\| \leq \epsilon_\omega$ or if $n > n_{\max}$. Go to step 2 if the stabilization test fails;
8. The final motion estimates are given by $\hat{\mathbf{t}}_n$ and $\hat{\omega}_n$.

As explained in section 3, the convergence of this algorithm depends on η_i. We propose to determine η_i in order to have small values of μ_{ij} for features with related motion out of the bound limits. From dead-reckoning, it is known that the real motion parameters belong to the validity intervals $\hat{\mathbf{t}} - \boldsymbol{\Delta}\mathbf{t} \leq \hat{\mathbf{t}} \leq \hat{\mathbf{t}} + \boldsymbol{\Delta}\mathbf{t}$ and $\hat{\omega} - \Delta\omega \leq \hat{\omega} \leq \hat{\omega} + \Delta\omega$. Thus, given the motion validity intervals and the feature $\mathbf{f}_i$, we must determine the maximum allowable value for d, called $d_{\max}$. In order to avoid a search procedure in the motion validity interval, we assume that such a maximum is on its limits. Hence,

$$d_{\max} = \left\|\boldsymbol{\lambda}(\mathbf{f}_i, \hat{\mathbf{t}}, \hat{\omega}) - \boldsymbol{\lambda}(\mathbf{f}_i, \hat{\mathbf{t}} + \boldsymbol{\Delta}\mathbf{t}, \hat{\omega} + \Delta\omega)\right\|. \quad (24)$$

Eq. (24) suggests that the real motion parameters are $\hat{\mathbf{t}}$ and $\hat{\omega}$, the feature corresponding to $\mathbf{f}_i$ is $\boldsymbol{\lambda}(\mathbf{f}_i, \hat{\mathbf{t}}, \hat{\omega})$, and the maximum error corresponds to $\hat{\mathbf{t}} + \boldsymbol{\Delta}\mathbf{t}$ and $\hat{\omega} + \Delta\omega$. We define η_i as

$$\eta_i = -\frac{d_{\max}^2}{\log 0.5}, \quad (25)$$

which means that $\mu_{ij} \leq 0.5$ for $d(\mathbf{f}_i, \mathbf{f}_j, \hat{\mathbf{t}}, \hat{\omega}) \geq d_{\max}$.

The algorithm presented above keeps the motion bounds $\boldsymbol{\Delta}\mathbf{t}$ and $\Delta\omega$ constant. It may be also interesting to update the motion bounds at each iteration n. Thus, we propose to update the motion bounds by the following rule:

$$\boldsymbol{\Delta}_n = \begin{cases} \boldsymbol{\Delta}_0 \exp(-n/\tau) & \text{if } n \leq n_s, \\ \boldsymbol{\Delta}_{n-1} & \text{elsewhere.} \end{cases} \quad (26)$$

In eq. (26), $\boldsymbol{\Delta}_n = \begin{pmatrix} \boldsymbol{\Delta}\mathbf{t}_n^T & \Delta\omega_n \end{pmatrix}^T$ represents the motion error bounds at n^{th} iteration, n_s is the maximum iteration number for which the error bounds are allowed to be updated. τ is a constant which must be determined as a function of the desired value of $\boldsymbol{\Delta}_n$ for $n > n_s$. It is interesting to have $n_s < n_{\max}$, otherwise the estimates may not stabilize before the $n_{\max}^{th}$ iteration.

5 Experimental Results

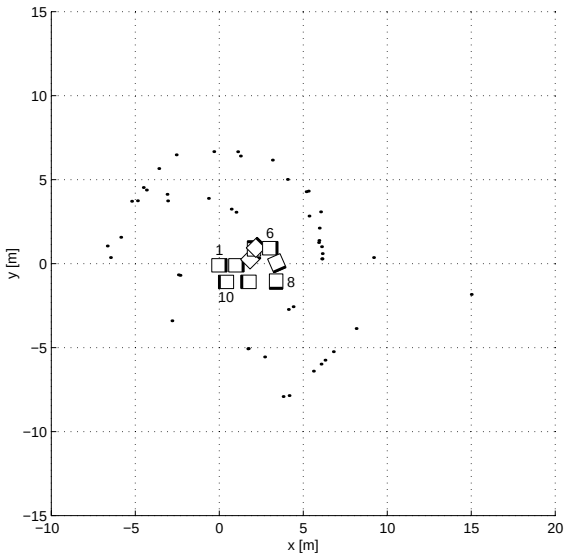

Figure 2: Superposed range images of experiment A.

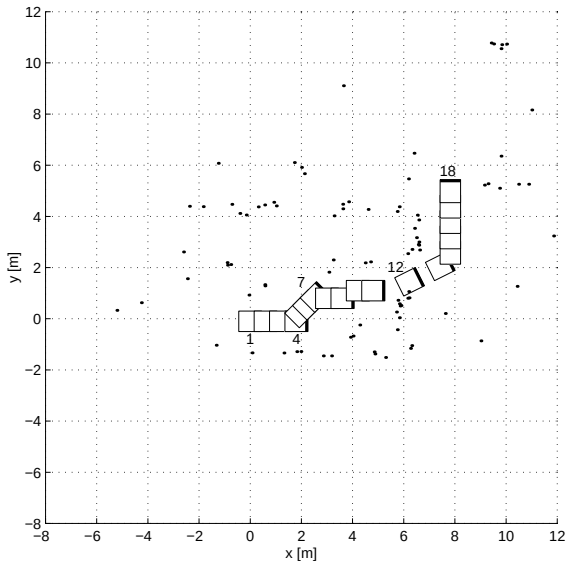

Figure 3: Superposed range images of experiment *B*.

The performance of the motion estimation algorithm was evaluated with two experiments in real indoor environments including moving obstacles. For each experiment, our omnidirectional robot Omni equiped with a laser radar was positioned at different places following a given trajectory. The corresponding absolute poses were carefully measured on the ground. These experiments are referred to as *A* and *B*. Figures 2 and 3 show the superposed range images for experiments *A* and *B*.

The laser radar, a *Ladar 2D30 IBEO Lasertechnik*, provides range images with an angular view of 270^o and is capable of detecting obstacles at distances up to 30 m. Its range imprecision was characterized as having a deterministic component which changes with the distance and a precision of ± 5 cm.

To evaluate the robustness of the proposed approach, the initial values for the motion estimates were simulated. The initial estimates are given by $\hat{\mathbf{t}}_0 = \mathbf{t} + \boldsymbol{\varepsilon}_\mathbf{t}$ and $\hat{\omega}_0 = \omega + \varepsilon_\omega$, where $\mathbf{t}$ and ω are the motion parameters measured on the ground and $\boldsymbol{\varepsilon}_\mathbf{t}$ and ε_ω are uniformly distributed random variables in the intervals $[-\boldsymbol{\Delta}\mathbf{t}_0, \boldsymbol{\Delta}\mathbf{t}_0]$ and $[-\Delta\omega_0, \Delta\omega_0]$. Considering the real motion magnitude, the initial error bounds were chosen as $\boldsymbol{\Delta}\mathbf{t}_0 = (0.3\ m, 0.3\ m)$ and $\Delta\omega_0 = 10^o$. We then have a greater uncertainty on the initial estimates than we would have had by dead-reckoning for the considered displacements. The algorithm was configured with $n_{\max} = 10$, $n_s = 5$, and τ was determined such that $\boldsymbol{\Delta}_{n_s} = (0.1\ m, 0.1\ m, 1^o)$. $\boldsymbol{\Delta}_{n_s}$ was empirically chosen considering the ladar imprecision.

For the line extraction procedure, we have developed an algorithm based on the fuzzy c-means clustering algorithm [8], but using a different representation for the linear prototypes. Unfortunately, given the limited space for this publication, we are not allowed to give further presentation of this algorithm. The segmentation algorithm extracted lines with $\sigma \leq 0.05$ m and a minimal length of 0.2 m. They include at least 5 scan points, with a maximal distance between two consecutive points of 0.5 m. Ellipsoidal clusters were selected from the extracted lines as having a maximum length of 0.5 m. The extracted lines with length greater than 0.5 m were selected as linear clusters. An example of extracted features is shown in Fig. 4. This figure shows how difficult it can be to interpret an image where partial occlusions occur.

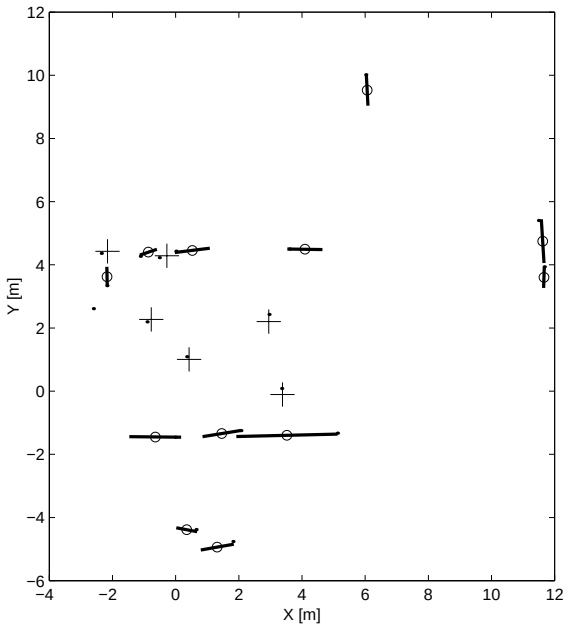

Figure 4: Segmentation example: linear clusters (-o-) and ellipsoidal clusters (+).

The motion estimation errors for experiment *A* are shown in Figure 5. As the algorithm estimates relative motion between poses P_{k-1} and P_k, the presented errors are ploted versus the index k. As performance measures, we use $\bar{\varepsilon}_{t_x}$, $\bar{\varepsilon}_{t_y}$ and $\bar{\varepsilon}_\omega$, the mean absolute errors in t_x , t_y and ω, respectively. For experiment *A*, $\bar{\varepsilon}_{t_x} = 2.89$ cm, $\bar{\varepsilon}_{t_y} = 4.65$ cm and $\bar{\varepsilon}_\omega = 1.13^o$. For images $I_8 - I_9$, the algorithm was ill-conditioned. As we can see in Figure 2, the rotation executed by the vehicle between these images was very important ($\omega = 90^o$). As a consequence, there were not enough corresponding features.

Figure 6 shows the motion estimation errors for experiment *B*. In this experiment, the robot moved

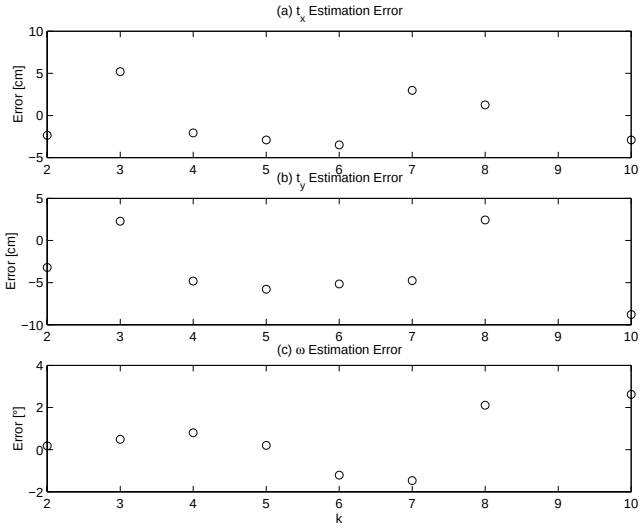

Figure 5: Motion estimation errors for experiment A.

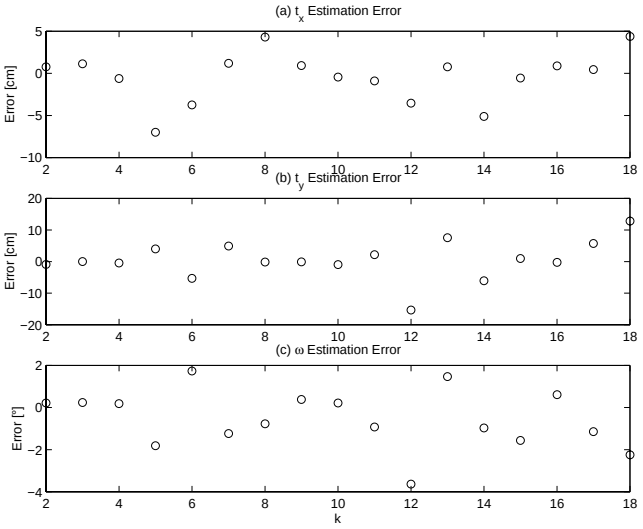

Figure 6: Motion estimation errors for experiment B.

across three different rooms. It is shown that the worst estimates were obtained for images I_{12} and I_{18}, when the robot changed room. Although the room changes, the estimates were not strongly affected. The performance measures for this experiment were $\bar{\varepsilon}_{t_x} = 2.15\ cm$, $\bar{\varepsilon}_{t_y} = 3.98\ cm$ and $\bar{\varepsilon}_{\omega} = 1.13^o$.

The estimation errors for these experiments are close to the ladar imprecision.

6 Conclusion

A new method for relative motion estimation of mobile robots from range images was presented in this article. The algorithm aims to match two consecutive range images in the feature domain. It iteratively evaluates a feature correspondence measure and calculates motion estimates in order to minimize a cost function. As a local matching approach, the proposed solution uses initial motion estimates and associated uncertainty provided by dead-reckoning. Experiments carried out in real indoor environments confirm the good performance of this approach.

Such a method may be also applied to solve the absolute localization problem with respect to a refence map. Moreover, we plan to extend the approach by incorporating a confidence measure for the 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] A. A. Holenstein, M. A. Müller, and E. Badreddin. Mobile robot localization in a structured environment clutterd with obstacles. In *Proc. of IEEE ICRA*, pages 2576–2581, May 1992.

[4] Feng Lu and Evangelos Milios. Robot pose estimation in unknown environments by matching 2-D range scans. *Journal of Intelligent and Robotic Systems*, 18(3):249–275, 1997.

[5] 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.

[6] Artur Arsenio and M. Isabel Ribeiro. Active range sensing for mobile robot localization. In *Proc. of IEEE/RSJ IROS*, October 1998.

[7] Rajesh N. Davé and Raghu Krishnapuram. Robust clustering methods: A unified view. *IEEE Transactions on Fuzzy Systems*, 5(2):270–293, 1997.

[8] J. C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms.* New York: Plenum, 1981.