Journal of Harbin Institute of Technology (New Series), Vol. 11, No. 4, 2004

A feature extraction and correspondence algorithm for laser range finder with sensor uncertainty

SUN Ying-jie[1], *CAO Qi-xin*[1], *LEE Jay*[2]

孙英杰, 曹其新, 李 杰

(1. Robotics Research Institute, Shanghai Jiao Tong University, Shanghai 20030, China, E-mail: {yjsun, qxcao} @sjtu.edu.cn;
2. Intelligent Maintenance Systems, University of Wisconsin, Milwaukee 53201, U.S.A, E-mail: jaylee @uwm.edu)

Abstract: This paper presents a feature extraction and correspondence algorithm which employs a novel feature transform. Unlike conventional approaches such as Hough Transform, we employ a robust but simple approach to extract the geometrical feature under real dynamic world conditions. Multi-threshold segmentation and the split-and-merge method are employed to interpret geometrical features such as edge, concave corners, convex corners, and segments in a unified framework. The features are represented by feature tree (F-Tree) so as to compactly represent the environments and some important properties of the F-Tree are discussed in this paper. To demonstrate the validity of the approach, we show the actual experiment results which are based on real Laser Range Finder data and real time analysis. The comparative study with Hough Transform shows the advantages and the high performance of the proposed algorithm.

Key words: feature extraction and correspondence; multi-threshold segmentation; eigenspace; feature tree; sensor uncertainty

CLC number: TP242.6 **Document code:** A **Article ID:** 1005-9113(2004)04-0361-07

The localization and navigation methods which operate on the basis of geometric reasoning are highly dependent on the reliable extraction and recognition of features from sensor data of an autonomous mobile robot[1]. A key research component of the SLAM (Simultaneous Localization and Map Building) problem is that of reliable feature detection and subsequent association. Image-based methods extract features based on edge formations, such as corners or straight lines[2], or perform segmentation on the basis of intensity or color, while sonar-based methods attempt to link sonar points into lines and structures[3].

A geometrical feature is a special type of target that can be reliably observed in successive sensor measurements and that can be accurately described in terms of a concise geometric parameterization[1]. Feature-based methods are concerned primarily with optimizing feature correspondence, and are susceptible to local minima in the function to be optimized, especially when employed with large-scale maps. Furthermore, these methods often rely on an accurate a priori map which is usually obtained from architectural drawings, or by manual measurement, which can fail to account for the presence of furnishings such as desks or chairs, or the issues of the dynamics of human and robot interaction with the environment.

A popular alternative to extracting naturally occurring features from sensor data is to employ artificial landmarks, that is, features which are not natural to a particular environment, but which are inserted, affixed, or otherwise deployed on the basis that they can be more robustly detected and extracted by a sensor. Artificial landmarks benefit from the ability to easily extract parameters based on a priori knowledge of landmark geometry, or through explicit labeling, such as bar codes or ultrasonic beacons[1,4]. The use of artificial landmarks can greatly simplify the problem of position estimation but there are significant drawbacks due to the fact that they require prior (and often human) intervention and can impose other costly or impractical requirements on the environment.

Employing a feature level description of the robotic environment, the two important issues of feature extraction and correspondence are addressed in this paper. The former is known as the detection problem, and the latter the registration (or correspondence) problem. In a dynamic changing environment, it is difficult to maintain robust feature extraction and correspondence. For most feature-based methods, the choice of which features to employ is often sensor dependent and constrained to a particular application domain. Popular algorithms for feature matching and model extraction fall into two broad categories: generate-and-test and Hough transform variations. However, both methods present problems in practical implementations.

Received: 2003-10-08.
Sponsored by the National Natural Science Foundation of China (Grant No. 50128504 and 50390063).

Generate-and-test methods are sensitive to noise in the data. They often fail when the generated model fit is poor due to error in the data used to generate the model position. Hough transform variations are less sensitive to noise, but implementations for complex problems suffer from large time and space requirements and from the detection of false positives[5]. Range-weighted Hough transform has been employed to extract directions and distances to the walls so as to navigate a mobile robot in cluttered rooms[6]. Although Hough Transform is an effective method for detecting lines and curves from noisy data, it has the drawback of sensitivity to the resolution of the discretization resolution[6].

Uncertainty plays an important role in mobile robot navigation at many levels, such as sensor interpretation, sensor fusion, map making, path planning, self-localization, and control[4]. Sensors are typically noisy and the information they provide is often ambiguous or misleading. For example, the sensor uncertainty of mobile robots prevents them from knowing their exact location during navigation. Due to the statistical error, navigation methods for mobile robots need to take various sources of uncertainty into account in order to get robust performance. Feature extraction and correspondence become difficult because of the inherent noise of almost all sensors which often leads to instability in the extracted features. In recent years Laser Range Finders have been used extensively for navigation in autonomous mobile robot systems due to their high resolution and reliable performance[2,7]. Unlike conventional approaches such as Hough Transform, we employ a robust but simple approach to extract the geometrical feature under real dynamic world conditions which provide the ability to cope with the sensor uncertainty.

This paper is organized as follows: Section 1 gives an overview of the Laser Range Finder and discusses its performance. The feature extraction algorithm, split-and-merge method and feature representation approach are discussed in Section 2. The experiment results are described in Section 3. Section 4 gives a discussion and concludes this paper.

1 Laser Range Finder (LRF)

Although most robots today still employ some form of sonar due to the cost and power consumption advantages, in our work LRF (See Fig. 1) is employed in the feature extraction due to its high performance. Firstly, the sonar data are sparse and many sonar devises have to be mounted around the robot while a laser scanner is a dense ranging device. Secondly, the response time of the LRF is much shorter than sonar. Finally, LRF data is very reliable and stable in contrast to sonar which has disadvantages of multiple specular reflections[8,9], slow processing speeds and wide beam width which gives rise to large angular uncertainty in measurement as mentioned above.

Fig. 1 Laser range finder LMS200

1.1 LMS 200 Range Data

A SICK LMS 200 Laser Range Finder is an optical sensor which scans its surroundings with infrared laser beams. As a result of this scanning principle, the LMS requires neither separate receivers nor reflectors. The sensor operates on the principle of reflex light time measurement. It emits very short light pulses. At the same time an electronic stopwatch is running. If the light encounters an object, it is reflected and thrown back to the sensor. From the time between sending and receiving (Δt), the sensor is able to calculate its distance from the object. In the sensor there is also a uniformly rotating mirror, which deflects the light pulses so that they sweep a semicircular area. By determining the mirror angle, the LMS detects the direction at which the object is located. The sensor will determine its precise position from the measured distance and the direction of the object.

As mentioned before, a laser range finder is equipped to measure the distances of objects in the environment. An object in the working space W is represented as an observed range sensor data set $O_i = (r_i, \theta_i)$, where r_i is the distance, θ_i is the angle relative to a predefined direction (e.g. X-axis), and i is a numbered index. In this paper, we assume that all the indices are enumerated in a counterclockwise direction (Fig. 2). $O(r_i, \theta_i)$, where $r_i \in R$, $R = [0, R_{\max}]$, $i \in I$, $I = [0, \theta_{\max}]$ are the polar coordinates of the points. The points are sequentially acquired by the laser range finder with a given angular resolution.

1.2 LRF Configuration

The LRF module controls the LRF and provides the raw sensor data. This sensor provides scans in an 180° angular field with a resolution of 1° through its RS232 interface at 19200 baud. That is to say $O(r_i, i)$, where $r_i \in R$, $R = [0, 8000]$, $i \in I$, $I = [0, 179]$ (See Fig. 2). The data are acquired within a period of about 200 ms. Its maximum range is set to be 8 000 mm, with a statistical error +/ - 15 mm.

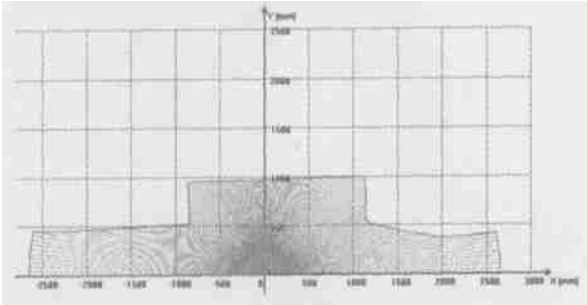

Fig. 2 Laser range finder range data

2 Feature Extraction and Correspondence Algorithm

2.1 Feature Transform

First we transform the polar coordinate $O(r_i, \theta_i)$ into the Cartesian coordinate system (x_i, y_i), where $x_i = r_i\cos\theta_i$, $y_i = r_i\sin\theta_i$ (Fig. 2). Next we calculate the forward deference: $\delta x_i^f = x_i - x_i + \eta$, $\delta y_i^f = y_i - y_i + \eta$ and backward deference: $\delta x_i^b = x_i - x_i - \eta$, $\delta y_i^b = y_i - y_i - \eta$ on (x_i, y_i). The range image is divided into intervals with step length η which is decided by the sensor uncertainty field. The Feature Transform is defined as follows: $\delta x_i' = \delta x_i^f - \delta x_i^b$, $\delta y_i' = \delta y_i^f - \delta y_i^b$.

2.2 Multi-threshold Segmentation and Feature Extraction

Once the original data is mapped into feature space, the eigenvalue of the intervals are calculated to determine different types of geometrical features, which are then used to register the features in the whole map. In order to improve the robustness of the feature transform and consider the sensor uncertainty implicitly, we use an observing window with k width. The eigenvalue in the sensor feature space is defined as $F_i = \sum_{j=i}^{i+k} \delta x_j' \cdot \delta y_j'$, where k is the width of the observing window.

The feature type of the interval is decided by the threshold and the precise position of the feature point is calculated through $E_i = \underset{i \in [j, j+k]}{\arg\max} |\delta x_i' \cdot \delta y_i'|$. In order to unify the feature extraction algorithm into a framework, we employ the multi-threshold method.

$$C\theta = \{ i \in I \mid E_i \wedge \delta y_i' \le 0 \wedge (F_i \ge T_{\text{Concave}}) \},$$

$$V\theta = \{ i \in I \mid E_i \wedge \delta y_i' \ge 0 \wedge (F_i \ge T_{\text{convex}}) \},$$

$$E\theta = \{ i \in I \mid E_i \wedge ((F_i \ge T_{\text{Edge}}) \vee (F_i \le -T_{\text{Edge}})) \},$$

$$S_b = \{ i \in I \mid (T_{\text{Concave}} \ge F_i \ge T_{\text{Segenment}}) \vee (-T_{\text{Segenment}} \ge F_i \ge -T_{\text{Concave}}) \},$$

$$S_e = S_b + k.$$

Where $C\theta$ is the index of the concave corner, $V\theta$ is the index of the convex corner, $E\theta$ is the index of the edge, and S_b and S_e are the beginning and ending indices of the segments. The above detailed geometrical features are essential for the geometrical reasoning.

2.3 Split-and-Merge Feature Extraction Algorithm

The splitting phase starts from (r_0, θ_0) taking the whole range image as a candidate and splitting the raw range data according to the step length η. After the Feature Transform and the multi-threshold segmentation, the interval's type is decided by the Feature Transform and its eigenvalue. Feature vector: $L_i(i = 0, 1, \cdots, n) = (\theta_{i\text{Begin}}, \theta_{i\text{End}}, F)$, where $\theta_{i\text{Begin}}$ and $\theta_{i\text{End}}$ are the beginning and ending indices of the feature respectively. $F \in (U, E, C, V, S)$ represents the feature type. Where U represents that feature of interval is unknown; E represents that feature of interval that is the Edge; C represents that feature of interval that is Concave; V represents that feature of interval that is Convex; and S represents that feature of the interval that is the Segment.

After the spilt loop we merge adjacent candidate intervals if they are homogenous and extract task-specific knowledge (corners, straight lines, edges, etc.). The concave and convex corners are extracted by multi-threshold segmentation and their positions are also calculated. At the same time, the property of the geometrical feature could be calculated such as the length and distance and orientation (ρ, α, l) of the segment. It merges segments until their eigenvalues are greater than a threshold. The detail algorithm is shown in Fig. 3.

```
Algorithm: Split-and-Merge()
{
0 Given LRF range data,start from(r0,θ0);
1 divide the image into [180/η] intervals ;
2 perform Feature Transform and calculate the eigenvalue;
3 extract feature using multi threshold segmentation;
4 if the interval's feature Fi = C then registry a concave corner;
5 if Fi = V then registry a convex corner;
6 if Fi = S ∧ Fi-1 ≠ S then begin a segment and record θbeginS ;
7 if Fi = S ∧ Fi-1 = S then merge the adjacent intervals;
8 if Fi = S ∧ Fi+1 ≠ S then end a segment and record θendS .
}
```

Fig. 3 Slpit-and-merge algorithm

2.4 Sensor Uncertainty and Step Length Selection

For the forward and backward difference, the step length η is adjusted according to the object's position and the relative sensor uncertainty field. Fig. 4 shows the results of an experiment conducted for the purposes of demonstrating the sensor uncertainty of LRF. Fifteen continuous epochs of sensor readings are recorded while the sensor and the objects are both fixed. Fig. 5 gives a model of the sensor uncertainty and demonstrates the relationship between the sensor uncertainty and the step length selection. Another problem is that a previous approach[10] suffers from the round off error because the data is too dense and the difference between the adjoining points i.e. $\delta r_i = r_i - r_{i+1}$ is too small comparing to r_i and r_{i+1}.

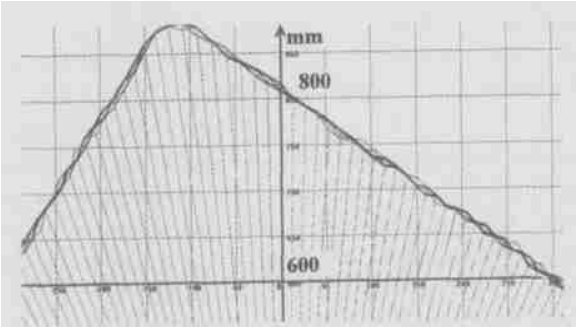

Fig. 4 LRF sersor uncertainty field

Considering the above problem and the sensor uncertainty, that approach[10] is not practical in this paper. If η is too small the round-off error will dominate the error and the if η is too large the truncation error will becomes big and reduces the resolution as illustrated in Fig. 5. Hough Transform has some inherent difficulties for the accurate segment extraction. Although a small quantization level of the angle could increase accuracy, the appropriate selection of the quantization step $\Delta\theta$ is necessary due to the false maxima of the accumulator matrix[11]. In order to improve the robust performance, the Hough Transform has to decrease the angle resolution of the segment extraction.

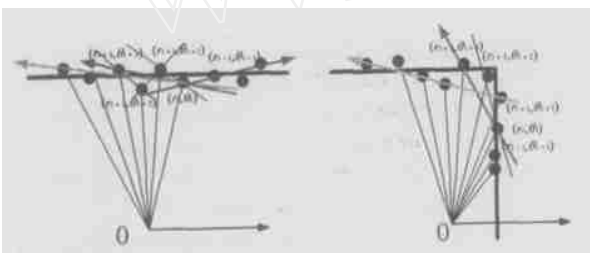

Fig. 5 Step length selection and sensor uncertainty

2.5 Feature Representations and Feature Tree

In the following we introduced the feature tree (F-tree). F-tree is used to represent the features which have been extracted. The F-tree enables the compact representation of environment and integrates the topological map building and incremental building of geometrical models for objects in a robotic context. The feature is not easily managed due to the occlusion. Feature tree is used to compactly represent the environment through a graceful data structure (Fig. 6). The feature tree is constructed sequentially and there are some important properties of the feature tree. In the following we will first give a detailed definition and then discuss the properties.

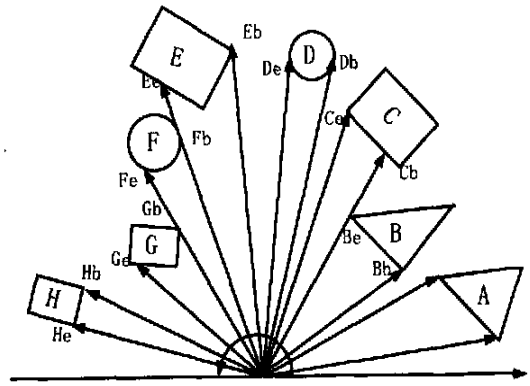

Fig. 6 Environment with cluttered objects

The F-Tree is incrementally constructed by extracting an object from the background and by getting the geometrical information $\delta r = r_i - r_{i-1}$, where $i \in I$. $O_{begin} = \{1 + i \in I \mid r_{i+1} - r_i < - T_{begin}\}$, $O_{end} = \{i \in I \mid r_{i+1} - r_i > T_{end}\}$, where O_{begin} and O_{end} are the beginning and ending indices of an object respectively.

Definition: Feature Tree is a data structure accessed beginning with the root node, the robot. Each node is either a leaf or an internal node. The left and right leaf nodes represent the beginning and the ending edges of the object respectively. The internal node stores the attributes of the object such as the feature type of the object or the length of the segment. All children of the same node are siblings (Fig. 7).

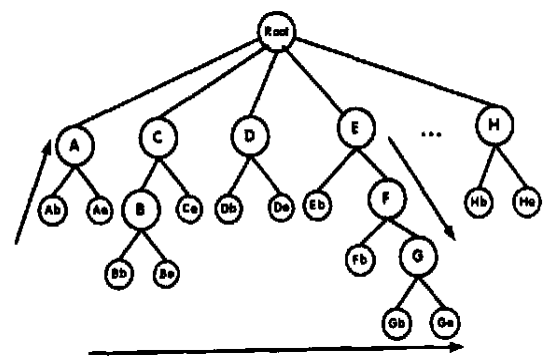

Fig. 7 Feature tree

The advantage of Feature Tree is that it considers the occlusion implicitly and sequentially builds an environment map. While it is evident that invisible objects cannot be tracked, objects that are occluded by another one passing between the range finder and the first object must be identified and presented correctly. Hence a Feature Tree should be able to cope at least with short occlusions. F-Tree represents the occlusion relationship compactly (Fig. 6). The property of the Feature Tree is discussed from the viewpoint of data structure.

Property 1 The depth, i.e. the distance from a leaf to the root of the feature tree, represents the occlu-

sion relationship of the objects. The deeper the tree is, the more difficult the occlusion relationship.

Property 2 The siblings represent the adjacent objects in the environment. See CD and DE in Fig. 7.

Property 3 If the left child of the internal node C is not a leaf node, the beginning edge of C is occluded by B. We temporarily take the end edge of B as the beginning edge of C and record that C is occluded by B (see BC in Fig. 7). If the right child of the internal node is not a leaf node, the property is similar (See EFG in Fig. 7).

Property 4 Feature Tree is an ordered binary tree. The key value of the leaf node is incremental from left to right following the arrow direction. This ordered tree makes traversing a feature tree to find a node with a given key very effective making feature management easy. In Fig. 7 we have the following relationships: Ab < Ae < Bb < Be = Cb < Ce < Db < De < Eb < Ee = Fb < Fe = Gb < Ge < Hb < He.

3 Experiment Result

3.1 Experiment Setup

The feature is determined by the eigenvalue after multi-threshold segmentation. The thresholds are decided by the relative sensor uncertainty field of the environment. In the experiments that follow we select the following parameters: $T_{\text{convex}} = T_{\text{Concave}} = 1.2 \times 10^3\ k$, $T_{\text{Segement}} = 500\ k$, $T_{\text{Edge}} = 2 \times 10^4\ k$. In this paper we let k be equal to the step length η. The distance from the object to the LRF is in the sensor field with a radius 3 m. Considering the tradeoff of the truncation error and the round off error, we select the step length as 3 to deal with the sensor uncertainty.

3.2 Feature Extraction and Correspondence Result

Figs. 8 and 9 illustrate that the feature representations in the Cartesian coordinate system are heterogeneous and there are several peaks in the above experiment results. The concave and convex corners cannot be distinguished uniquely from other features so that the geometrical feature cannot be extracted exactly from only one dimension such as $\delta x_j'$ or $\delta y_j'$.

If we perform the feature transform of x and y, i.e. $f(x, y) = \delta x_j' \cdot \delta y_j'$, from the feature correspondence results in Fig. 10, the concave corner, convex corner and segments are very obvious in the eigenspace and could be extracted and corresponded very easily as shown in Fig. 11. The Hough transform provides no connectivity information since it produces lines and not line segments. Consequently, a special structure has to be introduced in order to determine the end points of each line segment that is part of the same line[11]. The Hough Transform cannot extract the concave and convex corners directly while the feature correspondence results (see Figs. 10 and 11) validate that our algorithm can extract the geometrical features directly and represent them in a unified algorithm framework.

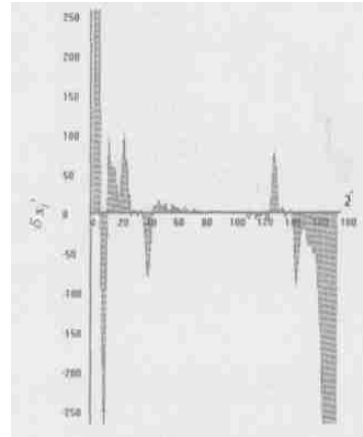

Fig. 8 Feature transform of x

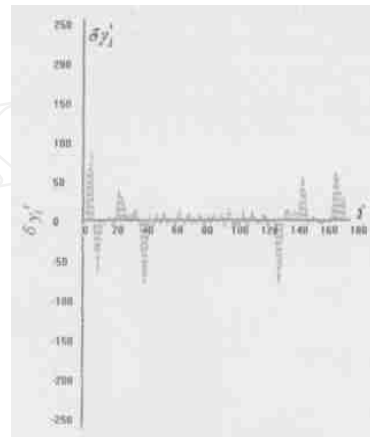

Fig. 9 Feature transform of y

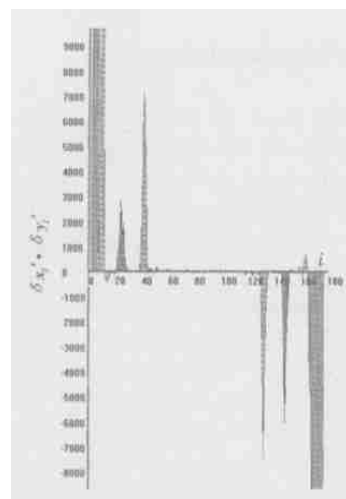

Fig. 10 Eigenspace and eigenvalue

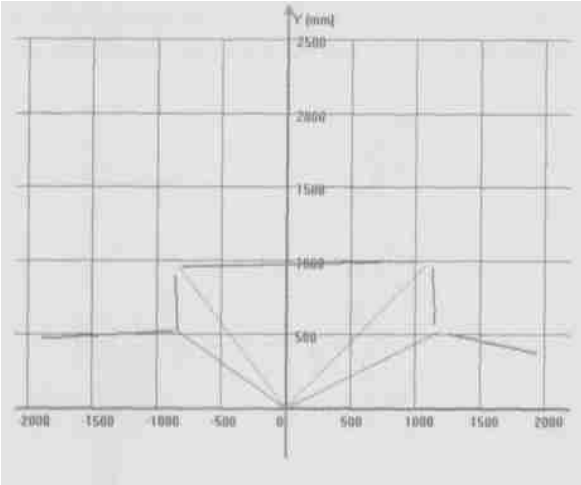

Fig. 11 Feature correspondence

3.3 Feature Management Result

The detected beginning and end edge of the objects are marked as bold triangles in Fig. 12. Fig. 12 shows the experiment results of constructing the feature tree, which matches with Fig. 7 very well. The results show the validity of using the feature tree to represent the occlusion relationship of the objects in the environment.

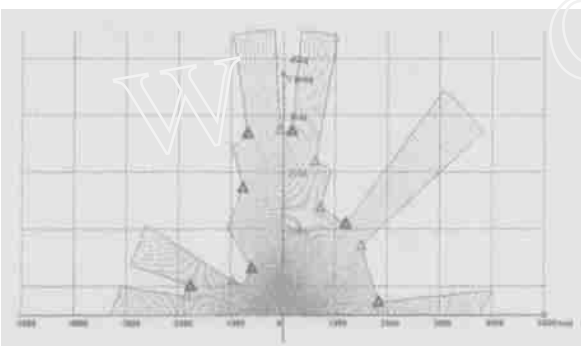

Fig. 12 Feature tree

4 Conclusions and Future Work

In this paper we presented a simple but robust feature extraction algorithm using the multi-threshold segmentation based split-and-merge method. The performance and the characteristics are studied from the experiment results. Further Feature Transform and feature eigenspace are defined in order to extract the geometrical features and represent the features in a unified framework. Feature extraction and correspondence is accomplished by learning a set of geometrical features called landmarks (i. e. convex corner, concave corner and straight line, etc.), each of which is detected as a local extremum of a measure of uniqueness and represented by an eigenvalue. After the original data is mapped into feature space, the eigenvalue of the intervals are calculated to determine different types of geometrical features which are then used to register the features in the whole map. In order to improve the robustness of the feature transform, the sensor uncertainty is discussed and considered implicitly.

F-Tree is introduced for the management of the geometrical feature of the environment and we presented a set of characteristics of the Feature Tree. The central problem in such systems is the robust performance of the feature extraction and correspondence algorithm, which integrates the sensor uncertainty into the adaptive step length selection. Experiment results obtained with the actual Laser Range Finder are presented and online analysis is conducted to show the feasibility and performance of the approach. We have implemented a reliable, robust and computationally efficient algorithm that uses Laser Range Finder to extract the geometrical features (i. e. natural landmarks) of the environment. The comparative study of feature selection methods of Hough Transform and our feature extraction algorithm shows that our algorithm can not only extract the concave and convex corners directly but also provides the connectivity information of the features. Once the rich features of the environment are extracted and managed, our work can be further extended into implementation of a robust and precise localization and navigation algorithm for an autonomous mobile robot.

Acknowledgement

The first author is grateful to Huang Libo and Luo Zhen for their valuable help.

References:

[1] LEONARD J, DURRANT-WHYTE H. Mobile robot localization by tracking geometric beacons[J]. IEEE Transactions on Robotics and Automation, 1991, 7(3): 376 - 382.

[2] ARRAS K O, TOMATIS N, JENSEN B T, *et al*. Multisensor on-the-fly localization: Precision and reliability for applications[J]. Robotics and Autonomous Systems, 2001, 34(2 - 3): 131 - 143.

[3] KIM J, PEARCE R A, AMATO N M. Robust geometric — based localization in indoor environments using sonar range sensors[J]. Proc IEEE Intelligent Robots and System, 2002, 1: 421 - 426.

[4] VLASSIS N, MOTOMURA Y, KR SE B. Supervised linear feature extraction for mobile robot localization [A]. Proc IEEE Int Conf on Robotics and Automation[C]. San Francisco: CA, 2000. 2979 - 2984.

[5] OLSON C F. A general method for geometric feature matching and model extraction[J]. International Journal of Computer Vision, 2001, 45(1): 39 - 54.

[6] FORSBERG J, LARSSON U, WERNERSSON A. Mobile robot navigation using the range-weighted hough transform [J]. IEEE Robotics and Automation Magazine, 1995, 2: 18 - 26.

[7] JENSFELT P, CHRISTENSEN H I. Pose tracking using laser scanning and minimalistic environmental models[J]. IEEE Transactions on Robotics and Automation, 2001, 17(2): 138 - 147.

[8] ARSHAN B, AYRULU B, UTETE S W. Neural network based target differentiation using sonar for robotics application

[J]. IEEE Trans. Robotics and Automation, 2001, 16(4): 435 - 442.

[9] JENSFELT P, AUSTIN D J, WIJK O, *et al*. Feature based CONDENSATION for mobile robot localization [A]. Proc IEEE Int Conf on Robotics and Automation[C]. San Francisco: CA, 2000. 2531 - 2537.

[10] ROUMELIOTIS S I, BEKEY G A. SEGMENTS: a layered, dual-Kalman filter algorithm for indoor feature extraction Proceedings[J]. IEEE/RSJ International Conference on Intelligent Robots and Systems, 2000, 1: 454 - 461.

[11] ANOUSAKI G C, KYRIAKOPOULOS K J. Simultaneous localization and map building for mobile robot navigation[J]. IEEE Robotics and Automation Magazine, 1999, 6(3): 42 - 53.