Topological Global Localization and Mapping with Fingerprints and Uncertainty

Adriana Tapus, Nicola Tomatis and Roland Siegwart

Autonomous Systems Lab
Swiss Federal Institute of Technology, Lausanne (EPFL)
1015 Lausanne, Switzerland
{Adriana.Tapus, Nicola.Tomatis, Roland.Siegwart}@epfl.ch

Abstract. Navigation in unknown or partially unknown environments remains one of the biggest challenges in today's mobile robotics. Environmental modeling, perception, localization and mapping are all needed for a successful approach. The contribution of this paper resides in the extension of the fingerprint concept (circular list of features around the robot) with uncertainty modeling, in order to improve localization and allow for automatic map building. The uncertainty is defined as the probability of a feature of being present in the environment when the robot perceives it. The whole approach is presented in details and viewed in a topological optic. Experimental results of the perception and localization capabilities with a mobile robot equipped with two 180° laser range finders and an omni-directional camera are reported.

1 Introduction

Navigation, described by Gallistel in [5] as the capacity to localize itself with respect to a map, is an elementary task that an autonomous mobile robot must carry out. Both, accurate perception and a reliable environmental modeling are needed, in order to localize a mobile robot and to build a map of its environment. Many methods have been proposed to represent environments in the framework of autonomous navigation, from precise geometric maps based on raw data or lines up to purely topological maps using symbolic descriptions. Each one of these methods is optimal concerning some characteristics but can be very disappointing with respect to other requirements. Metric maps are suited when the robot needs to know its location accurately in terms of metric coordinates. However, in office buildings with corridors and rooms, or roads, the topology of important locations and their connections might be sufficient for navigation. Topological maps are less complex and permit more efficient planning than metric maps. Moreover, it is easier to generate and maintain global consistency for topological maps than for metric maps. Even though research has recently leaded to successful solutions, robust perception for robot localization in unmodified, dynamic, real-world environments is still a challenge.

In this paper we concentrate on how multimodal perception combined with the uncertainty modeling of the features increases the reliability for topological localization and permits improving map building (map update). For the topological framework the fingerprint concept is used. This type of representation permits a reliable and distinctive environment modeling. The goal is to model the error-prone measurements from the imperfect exteroceptive sensors by means of uncertainty associated to their data.

Early works in topological localization [6] presented experiments in simulations, which avoided facing the perception problem. Following works as [10] were concerned with controlled environments, where perception with sonars was enough for the navigation purpose. Only more recent works address the perception problem in its whole complexity in the real world. Successful vision-based navigations are currently limited to indoor navigation because of its dependence on ceiling features [14], room geometry, or artificial landmark [12]. Other means for visual localization are applicable both indoors and outdoors, however they are designed to collect image statistics while foregoing recognition of specific scene features, or landmarks [13, 16]. In this context [7] and [8] introduced the fingerprint concept. Here, we show how the extension of the fingerprint concept with uncertainty modeling improves the topological global localization and mapping.

The remainder of this paper is organized as follows. We present in Section 2 the fingerprint concept, the way it is encoded and generated. In Section 3, we define the uncertainty model for the features present in the fingerprint. Section 4 is dedicated to the new method used for the fingerprint matching. In Section 5 a brief description of the localization and mapping is depicted. Experimental results are presented in Section 6. The system will use both, a laser scanner and an omni-directional camera for feature extraction. To conclude, Section 7 contains a discussion of the proposed approach and further research directions.

2 The Fingerprint Concept in a Topological Framework

The topological approach yields a compact representation and allows high-level symbolic reasoning for map building and navigation. With this method we try to eliminate the perceptual aliasing (i.e. distinct locations within the environment appearing identical to the robot’s sensors) and to improve the distinctiveness of places in the environment. To maximize the reliability in navigation, the information from all sensors available to the robot must be used. For this, the notion of fingerprint as described in [7, 8] is used. This characterization of the environment is especially interesting when used within topological localization and multiple sensor modalities.

2.1 Fingerprint Encoding

A fingerprint is a circular list of features, where the ordering of the set matches the relative ordering of the features around the robot. We denote the fingerprint sequence using a list of characters, where each character represents the instance of a specific feature type. In our case we choose to extract color patches and vertical edges from visual information and corners and beacons from laser scanner. We use the letter ′v′ to characterize an edge, the letters ′A′, ′B′, ′C′, ... , ′P′ to represent hue bins, the letter ′c′ to characterize a corner feature and the letter ′b′ to characterize a beacon feature. Details about the visual features extraction can be found in [7] and laser scanner features extraction in [2].

2.2 Fingerprint Generation

The fingerprint generation is performed in three steps (see Figure 1). The extraction of the different features (e.g. vertical edges, corners, color patches, beacons) from the sensors is the first phase of the fingerprint generation. The order of the features, given by their angular positions (0 … 360°) is kept in an array. At this stage a new type of feature, the virtual feature ′f′ is introduced. It reflects a correspondence between a corner and an edge. The ordering of the features in a fingerprint sequence is highly informative and for that reason the notion of angular distance between two consecutive features will be added. This adds geometric information and increases once again the distinctiveness between fingerprints. Furthermore, we introduced an additional type of feature, the empty space feature ′n′, for reflecting angular distances. Each ′n′ covers the same angle of the scene (20°). This insertion is the last step of the fingerprint generation. More details can be found in [8].

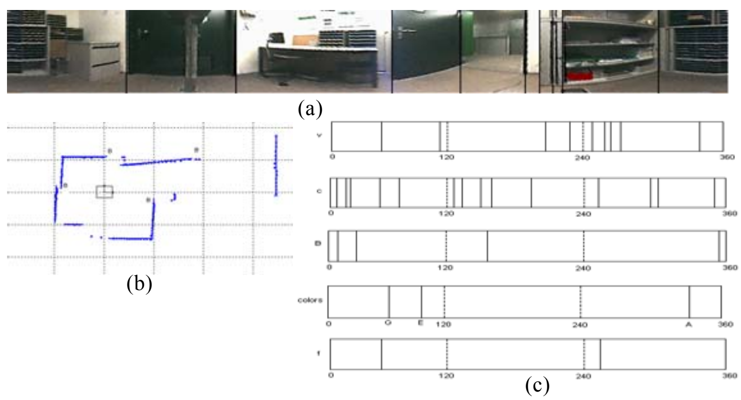

Figure 1. Fingerprint generation. (a) Panoramic image with the vertical edges and color patches (′v′ and color); (b) laser scan with extracted corners ′c′ and beacons ′b′; (c) images one to four depict the position (0 to 360°) of the vertical edges, the corners, the beacons and the colors (G-green, E-light green, and A-red). The fifth image describes the correspondence between the vertical edges and the corners. By regrouping all this together and by adding the empty space features, the final fingerprint is:
cbccbnfGcnEnvccncbcvncnnfvvvnccAcb.

3 Uncertainty Modeling in the Fingerprint Approach

The interaction between the mobile robot and its surroundings is performed by means of exteroceptive sensor data. Of course, the sensors are imperfect devices, and thus the measurements always contain errors. This can be modeled by associating uncertainty to their data. For that reason, the probability theory will be used to model the uncertainty of the geometric features extracted from the environment. We define the uncertainty as the probability of a feature of being present in the environment when the robot perceives it. The main idea is to introduce a new element in the fingerprint approach that specifies this uncertainty. Such uncertainty is modeled by experience, for each type of feature presented in Figure 1: vertical edges, colors, corners (extremities of the segments), beacons, ′f′ feature and respectively ′n′ feature. In the following, it will be shown how the uncertainty, denoted by the symbol *u*, is calculated for each one of the features:

- For the first three types of features (vertical edges, colors and corners) the uncertainty is calculated by using the following schema.

$$u = \begin{cases} \min, if\ \ extraction\_value \le low\_bound \\ linear\ interpolation, if\ \ low\_bound \le extraction\_value \le high\_bound \\ \max, if\ \ extraction\_value \ge high\_bound \end{cases}$$

  The *extraction_value* variable changes in function of the type of the feature. For the vertical edges, *extraction_value* corresponds to the gradient value. For the colors, the *extraction_value* is represented by the hue value of the color. In the case of the corner features, the *extraction_value* is identified as the distance between the robot and the extremities of the segments. The values of the *low_bound* and *high_bound* are experimentally determined for each type of feature. The *low_bound* represents the bound below which the feature has a low probability of existence so that the robot may not see it at the next passage. Another important element is the *high_bound*, above which a feature has a high certainty to exist and to be in the place where it was found (extracted). The *low_bound* and the *high_bound* are determined for each feature at the extraction level. The extraction of the vertical edges consists of the application of a threshold function on the gradient values. Since all edges below the threshold are ignored, the *low_bound* is used as the threshold value. The *high_bound* must be high, but not max_gradient, and so it has been fixed experimentally at *(threshold + mean_gradient)*. To extract the colors, a threshold function on the hue values has been applied and a similar method to that applied for the vertical edges has been chosen. Other two important elements in the schema described before are the values of *min* and *max*, which are fixed to 0.6, respectively 0.99. A method for calculating the value in-between these two values (*min* and *max*), by knowing the *low_bound*, *high_bound* and the *extraction_value* can be obtained by linear interpolation. In this case, *u* will be equal to:

$$u = \min + \frac{extraction\_value - low\_bound}{High\_bound - low\_bound} \times (\max - \min)$$

- The beacons are artificial landmarks (i.e. reflectors) and they are extracted with the help of the laser range finders. The experiments showed that the beacons are detected all the times, and for that reason the uncertainty has been fixed at a high value.
- As the ′f′ feature reflects the correspondence between a corner and an edge, its uncertainty is defined as the mean value between the uncertainty of the corner and the uncertainty of the vertical edge feature.
- The last feature is the ′n′ feature (i.e. the empty space feature that represents the angular distance between the features). The uncertainty of this feature is proportional to the distance between the features.

In this way, the uncertainty of the features used in the fingerprints is calculated. The limitation of this method resides in the models, which are difficult to define, especially for our definition of uncertainty, which cannot be directly derived from the physical characteristics of the sensors.

4 Fingerprint Matching

The string-matching problem is not easy. Usually strings do not match exactly because the robot may not be exactly located on a map point and/or some changes in the environment or perceptual errors occurred. Many string-matching algorithms can be found in the literature but they generally require the strings to have the same length. Some of them allow a level of mismatch, such as the k-mismatch matching algorithms and string matching with k differences [1, 3]. The first allows matches where up to k characters in the pattern do not match the text and the second requires that the pattern have an edit distance from the text of k or less. One of the main problems of the above methods is that they do not consider the nature of features and specific mismatches. We wish to consider the likelihood of specific types of mismatch errors. For instance confusing a red patch with a blue patch is more egregious than confusing the red patch with a yellow patch. The standard algorithms are quite sensitive to insertion and deletion errors, which cause the string lengths to vary significantly. The methods adopted previously in the fingerprint approach for sequence matching are the minimum energy algorithm used in stereovision [5] and the global alignment used usually for D.N.A. sequences [9]. Our current approach is an extension of the global alignment algorithm considering uncertainties and it is described below.

4.1 Global Alignment with Uncertainty

The global alignment algorithm finds an alignment between two strings so that the cost is minimal by using the cost function for aligning two characters.

Before starting describing the algorithm, the idea of aligning two strings and calculating the cost will be illustrated with an example (see Figure 2):

string1 := « abcd »
string2 := « bbc »

$$\text{cost}(x,y) := \begin{cases} x=\varepsilon \mid y=\varepsilon : & 0.6 \\ x = y & :0.0 \\ \text{else} & :1.0 \end{cases}$$

The cost of alignment $\begin{bmatrix} abcd \\ bbc\varepsilon \end{bmatrix}$ is calculated as:

$$= \text{cost}('a','b') + \text{cost}('b','b') + \text{cost}('c','c') + \text{cost}('d','\varepsilon')$$
$$1 \quad + \quad 0 \quad + \quad 0 \quad + \quad 0.6 \quad = \quad 1.6$$

Figure 2. An example of calculating the cost between two strings.

More formally, we can distinguish five elements, which form the global alignment algorithm (see Figure 3). The first element is an alphabet A, typically a set of letters, which is not empty. The second element corresponds to the two strings which are to be aligned: the first is composed of *m*, the second of *n* letters of the alphabet. The occlusion symbol is used to represent a space inserted into the string. The cost function gives the cost for the match between two symbols of the alphabet, included the occlusion symbol. Finally, the cost matrix is used to keep the minimal cost of a match between the first *i* letters of the first string with the first *j* letters of the second string, keeping this value in the element (*i*,*j*) of matrix V.

Alphabet	A, $A \neq \{\ \}$
Strings	$S1 \in A^m$, $S2 \in A^n$, $m, n \in N$
Occlusion symbol	ε, $\varepsilon \notin A$
Cost function	$f_{cost} : a \in A \cup \varepsilon, b \in A \cup \varepsilon \to \Re$
Cost Matrix	$V_{(i,j)} \in \Re$, $i \in \{0, 1, ..., m\}$, $j \in \{0, 1, ..., n\}$

Figure 3. The main elements of the Global Alignment algorithm.

The values of the cost function $f_{cost}(a, b)$, are calculated experimentally in function of the similarity between characters *a* and *b*, in other words the more similar the characters are, the lower will be the penalty for mismatching. It only remains to calculate the values of the elements of the cost matrix, which is constructed by a technique named "dynamic programming". Initially the edges of the matrix are initialized with the cumulative cost of occlusions. (That reflects the fact that we do not know, a priori, how much letters must be jumped in one or the other string in order to obtain the best solution.).
The base conditions of the algorithm are:

- $V(0, j) = \sum_{1 \le k \le j} f_{\cos t}(\varepsilon, S2(k))$
- $V(i,0) = \sum_{1 \le k \le i} f_{\cos t}(S1(k), \varepsilon)$

For *i* and *j* both strictly positive, the recurrence relation is:

$$V(i, j) = \min \begin{cases} V(i-1, j-1) + f_{\cos t}(S1(i), S2(j)) \\ V(i-1, j) + f_{\cos t}(S1(i), \varepsilon) \\ V(i, j-1) + f_{\cos t}(\varepsilon, S2(j)) \end{cases}$$

The three cases that can be distinguished from the above relation are:

- **Aligning** $S1(i)$ **with** $S2(j)$: The score in this case is the score of the operation $f_{cost}(S1(i),S2(j))$ plus the score of aligning i-1 elements of $S1$ with j-1 elements of $S2$, namely, $V(i\text{-}1,j\text{-}1)+ f_{cost}(S1(i),S2(j))$
- **Aligning** $S1(i)$ **with an occlusion symbol in string** $S2$: The score in this case is the score of the operation f_{cost} $(S1(i), \varepsilon)$ plus the score of aligning the previous i-1 elements of $S1$ with j elements of $S2$ (Since the occlusion is not an original character of $S2$), $V(i\text{-}1,j)+ f_{cost}$ $(S1(i), \varepsilon)$
- **Aligning** $S2(j)$ **with an occlusion symbol in string** $S1$: Similar to the previous case, the score will be $V(i,j\text{-}1)+ f_{cost}$ $(\varepsilon,S2(j))$.

If strings $S1$ and $S2$ are of length n and respectively m, then the value of their optimal alignment with the global alignment is the value of the cell (n,m).

The global alignment with uncertainty changes only the cost function described earlier. The cost function is adapted in order to take into account the corresponding uncertainty of features. The goal of adding the uncertainty in the string matching algorithm is to improve the distinctiveness of places. Next, a small example of global alignment algorithm with uncertainty will show the improvement of the matching (see Figure 4).

The example depicted in Figure 4 shows the improvement gained by the new fingerprint matching with uncertainty algorithm. Even if the two fingerprints from the map are similar (i.e. string1 and string2), the uncertainty of features will determine the map fingerprint that matches best the observed fingerprint (i.e. stringObs).

string1 := « abcde »
uncertainty_s1 := « 1 1 0.1 1 1 »
string2 := « abcde »
uncertainty_s2 := « 1 1 1 1 1 »

stringObs := « abde »
uncertainty_sObs := « 1 1 1 1 »

The alignment between the string1 and stringObs is the same with the alignment of the string2 and stringObs:

$$\begin{bmatrix} a\, b\, \varepsilon\, d\, e \\ ab\, c\, d\, e \end{bmatrix}$$

The probability that the stringObs matches the string1 is of 95% and the probability that the stringObs matches the string2 is of 75%.

Figure 4. An example of the Global Alignment algorithm with the uncertainty.

5 Topological Localization and Mapping

In this section a brief description of the global topological localization and map building approach is presented.

For the topological navigation a Partially Observable Markov Decision Process (POMDP) model is used.

A POMDP is defined as <*S,A,T,O*>, where *S* is a finite set of environment states; *A* is a finite set of actions; *T(s,a,s')* is a transition function between the environment states based on the action performed. A finite set *O* of possible observations and an observation function *OS* will be added. With this information, the probability of being in a state *s'* (belief state of *s'*) after having made observation *o*, while performing action *a*, is given by:

$$SE_{S'}(k+1) = \frac{OS(o,s',a)\sum_{s\in S} T(s,a,s')SE_S(k)}{P(o|a,SE(k))}$$

The key idea is to compute a discrete approximation of a probability distribution over all possible poses in the environment. An important feature of this localization technique is the ability to globally localize the robot within the environment. More details about this approach can be found in [4].

The information for the observation function within the topological framework is given by the fingerprint matching algorithm, described in the previous section.

While navigating in the environment, the robot firstly creates and then updates the global topological map. Each node contains the topology and door situation (i.e. corridor, T intersection, + Intersection, L Intersection, room, closed door, opened door, partially left opened door, partially right opened door and no door. For the doors, with the direction: in front, behind, on the left or on the right of the robot.) and the associated fingerprint. More details about the topology can be found in [12].

The entropy of a probability distribution p is

$$H(p) = -\sum_{s \in S} ps \log ps,$$

where $p_s \log p_s = 0$ when $p_s = 0$. The lower the value, the more certain the distribution. When the robot is "confused", the entropy is high. Therefore, the strategy of updating the map will be the following:

- When the entropy of the belief state is low enough, the map will be updated and so the fingerprint and the uncertainty of the features will also be updated.
- If the entropy is above a threshold α, then the updating will not be allowed, and we will try to reduce the entropy by continuing the navigation with localization.

Similarly to [15], when the robot feels confident concerning its state, it can decide if an extracted feature is new by comparing the observation fingerprint to the fingerprint from the map, corresponding to the most likely state. This can happen either in an unexplored portion of the environment, or in a known portion where new features appear due to the environmental dynamics. The features from the fingerprint come with their extraction uncertainty u. When a feature is re-observed, the uncertainty of the feature from the map fingerprint is averaged with the uncertainty of the extracted one. Otherwise, if the robot does not see an expected feature the uncertainty is decreasing. When the uncertainty of a feature from a map fingerprint is below a minimum threshold, than the feature is deleted, allowing in this way for dynamics in the environment.

6 Experimental Results

The approach has been tested in a 50 x 25 m^2 portion of our institute building.

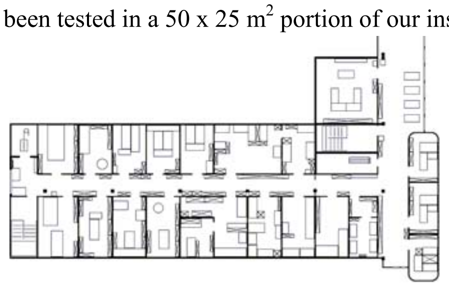

Figure 5. The test environment, with the rooms and corridors in which the experimentation has been done.

For the experiments, the Donald Duck robot (see Figure 6), a fully autonomous mobile robot, has been used.

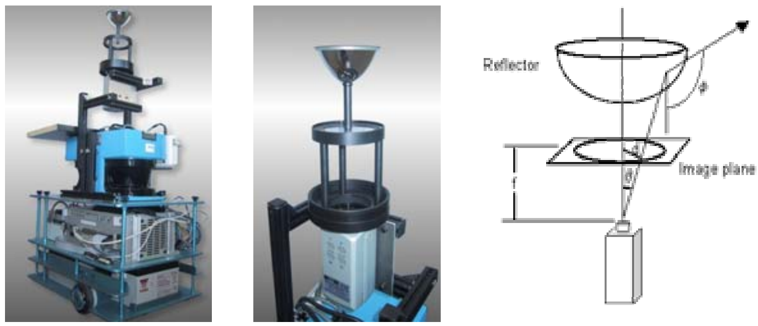

Figure 6. System used for experimentation: The fully autonomous robot Donald Duck and the panoramic vision system. The camera has a 640 x 480 pixel resolution and an equiangular mirror is used so that each pixel in the image covers the same view angle.

Its controller consists of a VME standard backplane with a Motorola PowerPC 604 microprocessor clocked at 300 MHz and running XO/2, a hard real-time operating system. Among its peripheral devices, the most important are the wheel encoders, two 180° laser range finders and an omni-directional camera. The panoramic vision system depicted in Figure 6 uses a mirror-camera system to image 360° in azimuth and up to 110° in elevation.

In order to validate the fingerprint approach with uncertainty, a comparison between the results obtained with the non-probabilistic approaches and the results obtained with the new probabilistic version will be presented. The experiments for all the approaches have been tested in the same environment and under the same conditions.

The test setup was the following: The robot extracted the four features (i.e. vertical edges, colors, corners and beacons) in seven offices at 11 different places. For the new matching approach, the uncertainties of different features have been modeled. One fingerprint per room has been included in a database as reference (map initialization) for the localization approach. The other 70 fingerprints have been matched to the database for testing the localization.

During all measurements, the orientation of the robot was approximately the same. This simplification could be omitted by letting the robot estimate his orientation by considering all rotations of the fingerprint string.

For a given observation (fingerprint) a match is successful if the best match with the database corresponds to the correct room. Table 1 illustrates the percentage of successful matching and the mean rank for three string matching algorithms: minimum energy, global alignment and global alignment with uncertainty. The rank calculates the position of the correct room, with respect to the others, in the classification (e.g. if the match is successful than the rank is 1, if the correct room is detected with the second highest probability the rank is fixed at 2, etc.).

Table 1. Classification using string matching, comparing minimum energy, global alignment and global alignment with uncertainty algorithms.

	right classifications	mean rank
minimum energy	58.82%	1.85
global alignment	75%	1.32
global alignment with uncertainty	83.82%	1.23

In Table 1, one can see the improvement from using global alignment with uncertainty instead of the global alignment or minimum energy algorithm. The results with global alignment with uncertainty algorithm have 83.82% of successful matches, which corresponds to a clear improvement of 8.82% with respect to the standard global alignment (see Table 1). Note that the experimental setup does not include yet the presented Partial Observable Markov Decision Process (POMDP) for localization. However, as soon as the matching information will be integrated by the POMDP, the motion will bring additional information to the system that should allow very reliably navigation.

7 Conclusion and Future Work

This paper has presented a method for topological global localization and mapping by using the fingerprint concept combined with an uncertainty modeling. The fingerprint approach [7, 8] has already shown its capability of representing real world scenes in a robust and flexible manner. The uncertainty model, presented here as the probability of a feature being present in the environment when the robot perceives it, improves the concept and allows for a more stable global localization. The performance of the probabilistic fingerprint approach is shown through experiments comparing the new approach with two old versions. From the experiments we can conclude that the presented method is practical and robust. The successful classification is 83.82% which represents an improvement of 25% in comparison with the minimum energy approach and 8.82% with the standard global alignment. Even though the matching is not yet integrated with POMDP, we can already state that all matching steps will bring important information to the system since the correct fingerprint has a mean rank of 1.23. Future work will focus on the integration of the fingerprint with uncertainty within a POMDP for Simultaneous Localization and Mapping (SLAM) and the extension of the whole approach towards multi-resolution SLAM.

Acknowledgments

This work was sponsored by the European project BIBA IST-2001-32115.

References

[1] Alfred V. Aho. Algorithms for finding patterns in strings. *In J. van Leeuwen, editor, Handbook of Theoretical Computer Science*, chapter 5, pages 254-300. Elsevier Science

Publishers B. V., 1990.
[2] Arras, K.O. and Siegwart, R, Feature Extraction and Scene Interpretation for Map-Based Navigation and Map Building, In Proceedings of the Symposium on Intelligent Systems and Advanced Manufacturing, Pittsburgh, USA, October 13-17,1997.
[3] Baeza-Yates, R., Navarro, G. Faster Approximate String Matching, Department of Computer Science, University of Chile, Santiago.
[4] Cassandra, A. R., L. P. Kaelbling, et al. (1996). Acting under Uncertainty: Discrete Bayesian Models for Mobile-Robot Navigation**.** *IEEE International Conference on Robotics and Automation*, Osaka, Japan.
[5] Gallistel, R. (1990), The Organization of Learning, MIT Press: Cambridge, MA.
[6] Kuipers, B. J. and Y. T. Byun (1987). A qualitative approach to robot exploration and map-learning. Workshop on Spatial Reasoning and Multi-Sensor Fusion, Los Altos, CA, USA, Morgan Kaufmann.
[7] Lamon, P., I. Nourbakhsh, B. Jensen and R. Siegwart (2001). Deriving and Matching Image Fingerprint Sequences for Mobile Robot Localization. IEEE International Conference on Robotics and Automation, Seoul, Korea.
[8] Lamon, P., Tapus A., Glauser E., Tomatis N., Siegwart R. (2003), Environmental Modeling with Fingerprint Sequences for Topological Global Localization, In Proceedings of the IEEE International Conference on Intelligent Robot and Systems, Las Vegas, USA, October 27-30.
[9] Needleman, S. and Wunsch, C. (1970), A general method applicable to the search for similarities in the amino acid sequence of two proteins, J. Molecular Biology, 48:443-453.
[10]Nourbakhsh, I. (1998). Dervish: An Office-Navigating Robot. Artificial Intelligence and Mobile Robots. D. Kortenkamp, R. P. Bonasso and R. Murphy, The AAAI Press/ The MIT Press: 73-90.
[11]Nourbakhsh, I., J. Bodenage, et al. (1999). An Affective Mobile Robot Educator with a Full-Time Job. Artificial Intelligence 114(1-2): 95-124.
[12]Tapus A., Ramel G., Dobler L. and Siegwart R. (2004), Topology Learning and Recognition using Bayesian Programming for Mobile Robot Navigation, In Proceedings of the IEEE International Conference on Intelligent Robot and Systems, Sendai, Japan.
[13]Thrun, S., Finding Landmarks for Mobile Robot Navigation, IEEE International Conf. on Robotics and Automation, May 1998.
[14]Thrun, S., Bennewitz, M., Burgard, W., Cremers, A.B., Dellaert, F., Fox, D., Hahnel, D., Rosenberg, C., Roy, N., Schulte, J., Schulz, D., MINERVA: a second-generation museum tour-guide robot, IEEE International Conf. on Robotics and Automation (Cat. No.99CH36288C) 1999.
[15]Tomatis, N., I. Nourbakhsh, and R. Siegwart (2003). Hybrid simultaneous localization and map building: a natural integration of topological and metric. *Robotics and Autonomous Systems,* 44:3-14**.**
[16]Ulrich, I. and I. Nourbakhsh (2000). Appearance-Based Place Recognition for Topological Localization. IEEE International Conference on Robotics and Automation, San Francisco, CA.References