Environmental Modeling with Fingerprint Sequences for Topological Global Localization

Pierre Lamon, Adriana Tapus, Etienne Glauser, Nicola Tomatis and Roland Siegwart

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

Abstract

In this paper a perception approach allowing for high distinctiveness is presented. The method works in accordance to the fingerprint concept. Such representation allows using a very flexible matching approach based on the minimum energy algorithm. The whole extraction and matching approach is presented in details and viewed in a topological optic, where the matching result can directly be used as observation function for a topological localization approach. The experimentation section will validate the fingerprint approach and present different set of experiments in order to explain practically the choice of different types of features.

1. Introduction and motivation

Research in mobile robot navigation has to focus on various issues in order to build a coherent working framework for navigation. Environmental modeling, perception, localization and mapping are all needed for a successful approach. Even though research has recently leaded to successful solutions, robust perception for the localization of a robot in unmodified, dynamic, real-world environments are rarely presented.

Current research has diverged to two different approaches:

- Metric: robot position defined by $\begin{bmatrix} x & y & q \end{bmatrix}^T$.
- Topologic: position defined by places or locations

In this paper we will concentrate in the perception and environmental modeling within a topological context. The robot will have to recognize its neighborhood in order to track its own position.

Early works in topological localization [10] presented experimentations in simulations, avoiding facing the perception problem. Following works [7] were concerned with controlled environments, where the perception with sonars was enough for the navigation purpose. Only recent works within the topological community address the perception problem in its whole complexity in the real world.

Successful vision-based navigation are currently limited to indoor navigation because of its dependence on ceiling features [1, 13], room geometry, or artificial landmark placement [8]. 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 [12, 15]. In this context [11] already introduced the fingerprint concept, but its perception was restricted to the CCD camera and the method has been studied in an absolute positioning approach, not a topological one.

The presented method will propose an approach for the representation of a location allowing for robust perception for the topological paradigm. The system will use both a laser scanner and an omnidirectional camera for feature extraction. The experimentation will focus on some important characteristics needed by a topological approach, like uniqueness and distinctiveness.

2. The fingerprint sequence

As the fingerprints of a person are unique, so each location has its own unique characteristics. Of course, when relying on the limited perceptual capabilities of a machine, it is difficult to guarantee the unique distinction between two similar places. This localization system assumes that a virtual *fingerprint* of the current location can be created and that the sequence generation methods can be made insensitive to small changes in robot position. If fingerprints are associated to each location, then the actual location of a mobile robot may be recovered by constructing a fingerprint and comparing it to its database of known fingerprints. However, this characterization of the environment is especially interesting when used within a topological localization framework. In this case the distinctiveness of the observed location plays an important role for the correct track of the robot position.

2.1 Fingerprint sequence encoding

We propose to create a fingerprint by assuming that a set of feature extractors can identify significant features in the environment around the robot. Omnidirectional sensors are preferred because the orientation as well as the position of the robot may not be known *a priori*.

We define a fingerprint as a circular list of features, where the ordering of the set matches the relative ordering of the features around the robot. In order to encode efficiently this circular list, we denote the fingerprint sequence using a list of characters, where each character represents the instance of a specific feature type. A similar representation can be found in [3]. Any number of feature detectors can be used. For example, if we choose to extract color patches and vertical edges from visual information we may use the letter 'v' to characterize an edge and the letters A,B,C,...,P to represent hue bins. We will use this example for illustrating the sequence-matching algorithm.

2.2 Fingerprints matching for localization

To introduce the problem of string matching, let us consider the example below. The first string has been extracted from the current location of the robot and the next two strings are strings from the database.

Place x: vvBEvvCvvvMvOBvvvvv

Database place 1: vvBEvMvCvvvMvMOBvvvv

Database place 2: LvLvvvBvvOLvBEvOvvv

Figure 1: Example of extracted fingerprints

As one can see the first string does not match exactly either of the others because the robot is not exactly located on a map point and/or some change in the environment occurred. Now what sequence match scoring method should we use to determine that the match is Place1 in this case and not Place2 with high confidence?

Great 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 k-mismatch matching algorithms, and string matching with k differences [2, 4]. Another approach for matching consists in considering strings as digital signals and computing the correlation. A measure of similarity will be in this case the height of the maximum peak of the correlation function.

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. Furthermore the standard algorithms are quite sensitive to insertion and deletion errors which cause the string lengths to vary significantly.

Minimum energy algorithm

The approach we have adopted for sequence matching is inspired by the minimum energy algorithm used in stereo-vision for finding pixels in two images that correspond to the same point of a scene [9]. As in the minimum energy case, the problem can be seen as an optimization problem, where the goal is to find the path that spends the minimum energy to go from the beginning to the end of the first sequence considering the values of the second one. The similarity between two sequences is given by the resulting minimum energy of traversal. Value 0 is used to describe a perfect match (e.g. self-similarity).

We describe our sequence matching algorithm using an example consisting of two particular sequences: "EvHBvKvGA" (length n = 9) and "EBCAvKKv" (length m = 8).

Initialization

First the initial n x m matrix must be built. The characters of the first string represent the rows and those of the second string the columns. To initialize this matrix only two parameters are needed. The first parameter is a number that represents the maximum mismatch value and the second is used to fix the minimum mismatch value between two different colors. In this particular example Max_init = 20 and Min_col = 5.

Init	E	B	C	A	v	K	K	v
E	**0**	11	8	14	20	20	20	20
v	20	**20**	20	20	0	20	20	0
H	11	20	**17**	17	20	11	11	20
B	11	0	5	**5**	20	11	11	20
v	20	20	20	20	**0**	20	20	0
K	20	11	14	8	20	**0**	0	20
v	20	20	20	20	0	20	**20**	0
G	8	17	14	20	20	14	14	**20**
A	14	5	8	0	20	8	8	20

Figure 2 : Init matrix. It represents the level of mismatch between the features.

If the corresponding features are of wholly different types (e.g. a color and an edge) then the corresponding matrix element is initialized to Max_init. If both features are vertical edges or represent exactly the same color the value 0 is used to describe a perfect match. If the comparison is between two colors, then the error is calculated according to the hue distance between the two colors, adjusted to inhabit the range from Min_col to Max_init.

Although a type-mismatch can be generally assigned a score of Max_Init, any newly introduced feature type must not only include the appropriate feature detector but also a mismatch table, identifying the score for various feature value comparisons within that feature type. This is an important aspect of the present work.

The cost matrix computation

This step computes the cost function for each cell of the matrix using only two parameters: the slope penalty (Slope_pen = 10) and the occlusion penalty (Occ_pen = 24). The cost for going from one cell to another depends on the initial value (see Fig. 2) of the target cell, the distance between the cells (the slope) and the cost value of the origin cell. Occ_pen is used for horizontal and vertical occlusion (slopes 0 and infinity respectively)

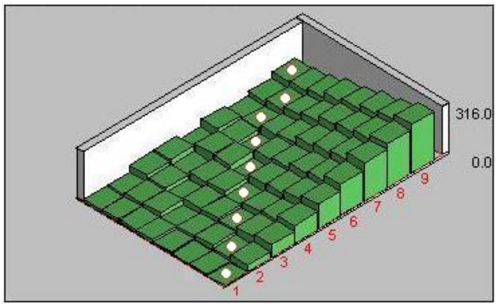


Figure 3: The Cost matrix and the optimal path (dots). There is an occlusion between the 7th and 8th line.

The best path

The minimum value of the last line of the Cost matrix corresponds inversely to the similarity between the two input sequences. In order to normalize the result this value is then divided by the worst value that can be obtained with two strings of similar length (in this case, result of the match between a string composed of m edges and one with n colors).

3. Perception and feature extraction

In this work the fingerprints are used for global localization in an office environment: the goal being to determine the office in which the robot is entering. The fingerprints can be very different depending on the position in the room. Even though the optimal solution would be to have no position requirements, in this work a positioning procedure ensuring that the robot will be close to the position on which the fingerprint's database has been extracted is adopted.

3.1 Positioning

There are many alternatives for the positioning of a robot in a room. The robot could detect the threshold of the door and stop at the entrance or enter in the room until a pre-defined distance has been reached. The drawback of these methods is that they don't consider the dimensions and the shape of the room. This can lead to non-optimal positions such as places close to obstacles obtruding most of the field of view of the room. Therefore, we assume that the position in the room with the most free space around it is the one with the highest probability of extracting numerous and characteristic features. This will ensure high distinctiveness of the observation.

The laser scanner is used to compute the gravity center of the free space. Because the angle between two scan points is constant, the density of points is inversely proportional to the distance. To account for that effect, the points must be weighted, giving the following equation for computing the gravity center:

$$\vec{x} = \frac{1}{n}\left[\sum_i w_i \cdot x_i \quad \sum_i w_i \cdot y_i\right]^T \qquad (1)$$

where n is the number of scan points and w_i the corresponding weights which are set equal to the distances r_i to the robot.

3.2 Visual feature extraction

Several types of features can be extracted from visual information. The color patches may be used but they are very sensitive to illumination and therefore their extraction suffers from a lack of robustness. We finally focused on edges because they are of particular value in structured environments such as indoor office buildings and are less sensitive to illumination changes. We have chosen to concentrate on vertical edges because of the instability and rarity of horizontal edges due to projection effects.

We proceed in two stages for extracting the edges. The first step consists in unwrapping the panoramic image. This process is computationally expensive but produces a rectangular image (1257x190) that is easier to handle with standard image processing algorithms. Furthermore, vertical edge detection is difficult and time consuming on raw panoramic images (see Fig. 8). The second step uses the same histogram-based method as the one presented in [11] for extracting the vertical edges. We choose to use the letter 'v' to characterize a vertical edge feature.

3.3 Laser scanner features extraction

Corners typically describe the shape of furniture, walls and other rectilinear obstacles. These features are semantically equivalent to the verticals extracted from the image. However, in contrast to the vertical edges, which are extracted with vision, corners are extracted with the laser scanner and are therefore only detected at a given height which corresponds to the position of SICK sensor on the robot.

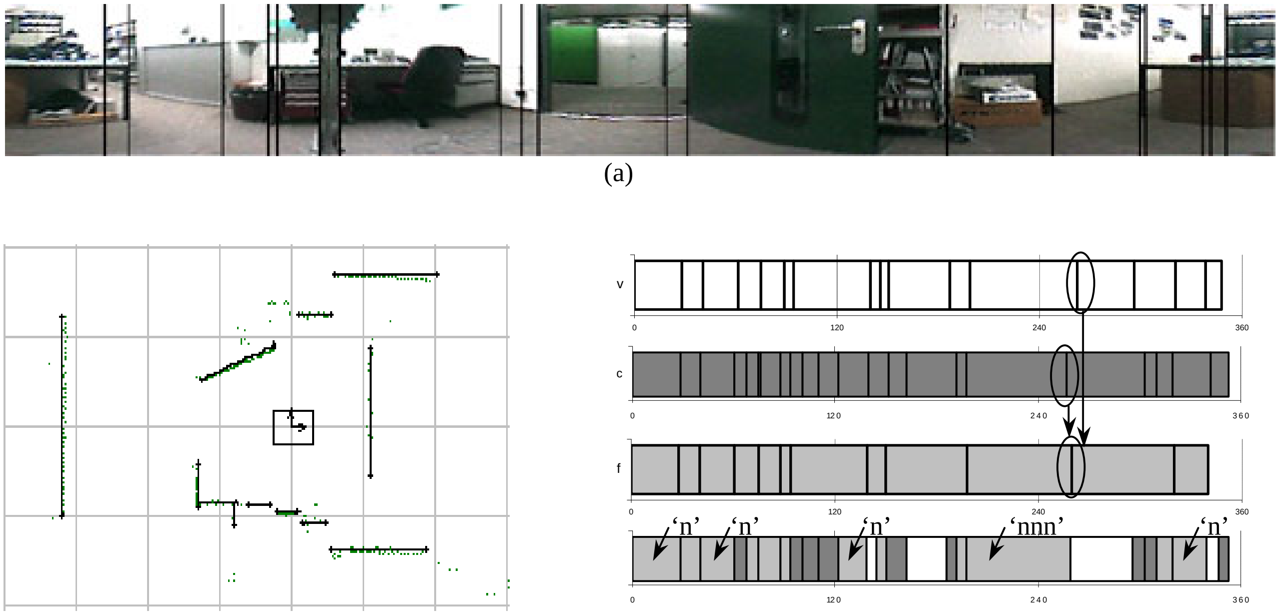
In order to detect the corners, we carry out an extraction of line segments, as illustrated in figure 4b. The Hough transform is then used [6]. The algorithm computes the slope between each successive point to find out which consecutive points belong to the same physical segment. This method presents the advantage of filtering the noise and all the dubious points. Finally, each end of the line is kept and regarded as an angle. We choose to use the letter ‘c’ to characterize a corner feature.

4. Fingerprint generation

The fingerprint extraction is performed in three steps (see Fig. 4). The first step consists in extracting the vertical edges from the panoramic image and the corners from the laser scan. The features are arranged in an array along with their corresponding position (from 0 to 359 degrees).

At this stage we introduce a new type of feature that reflects a correspondence between a corner and an edge: the feature ‘f’. This is natural since this correspondence effectively describes a third feature type. For example, a black line on a wall will generate a vertical edge but won’t be detected by the laser scanner. Inversely, a corner at the intersection of two white walls won’t necessarily be detected as a vertical edge. The third possibility is that the same element in the scene is detected by the two sensors. The addition of that type of feature increases the distinctiveness of the generated fingerprint. If two consecutive ‘v’ and ‘c’ features are close enough (5 degrees) they will be fused into an ‘f’ feature: this is the second step of the extraction.

The ordering of the features in a fingerprint sequence is highly informative. However, introducing the notion of angular-distance between two consecutive features can be of high interest. This adds geometric information and increases once again the distinctiveness between the fingerprints. We found important to add this information without changing the structure of the fingerprint and the matching algorithm. Therefore, we decided to introduce a new type of feature, the empty space feature ‘n’, for reflecting angular distance. Each ‘n’ covers the same angle of the scene (20 degrees). So, as many empty spaces as needed must be inserted in order to fill the gap between two consecutive features. This insertion is the last step of the fingerprint generation method.



(b)

(c)

Figure 4 : Fingerprint generation. (a) vertical edges detection ‘v’ (b) laser scan with extracted corners ‘c’ (c) the first two graphs depict the positions (0 to 360°) of the vertical edges and the corners respectiveley. Graph ‘f’ shows the result of the correspondance between the features ‘v’ and ‘c’. The last graph depict the fingerprint before the insertion of the empty space features(the arrows show the insertion points for the empty spaces features). The final string is : n f f n f c f f f c c c f v f c n v c f n n n f n v c c f n f v c

5. **Experimental Results**

The approach has been tested in the 50 x 25 m^2 portion of the institute building shown in figure 5.

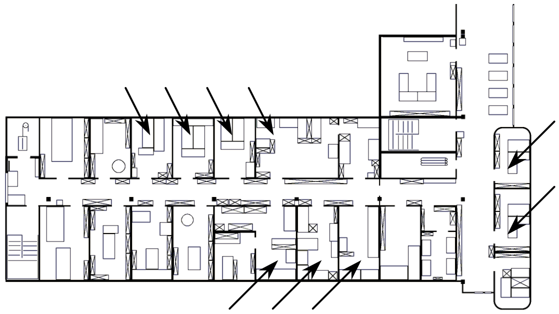


Figure 5: The test environment. The arrows indicate the rooms in which the experimentation has been done. The tenth room is not represented on the image.

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

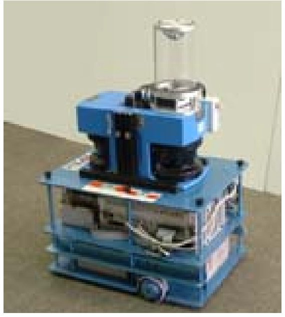


Figure 6: The fully autonomous robot Donald Duck. The panoramic vision system has been mounted right above the wheels. So, for a given position, the fingerprint extraction does not depend on the robot's orientation.

Its controller consists of a VME standard backplane with a Motorola PowerPC 604 microprocessor clocked at 300 MHz running XO/2, a hard real-time operating system. Among its peripheral devices, the most important are the wheel encoders, a 360° laser range finder and an omnidirectional camera. The panoramic vision system depicted in figure 7 uses a mirror-camera system to image 360° in azimuth and up to 110° in elevation.

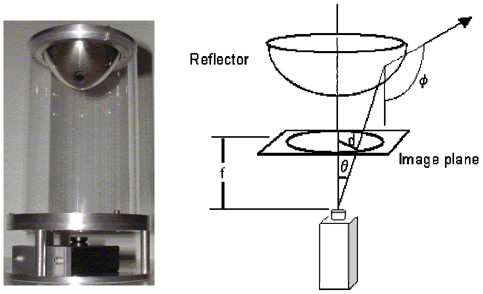


Figure 7: The panoramic vision system. The camera has a 640x480 pixels resolution and an equiangular mirror is used so that each pixel in the image covers the same view angle.

The use of an omnidirectional camera combines the advantages of the SICK laser range finder (e.g. an angle of view of 360°) and the capability of detecting verticals. This will bring considerable information to the system.

Positioning

In this section the positioning capabilities of the robot is tested and analyzed. For the experiments, the robot visited ten rooms of the environment depicted in figure 5 and tested the method four times in each room. The results are conclusive: the robot reaches all the time the center of the free space (see Fig. 8).

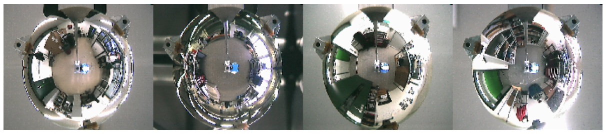


Figure 8: Different views of the free space taken by the panoramic vision system. The position of the robot corresponds to the center of the free space measured by the laser scanner.

The repeatability was also tested: the robot reached an area of radius of 15 cm around the center of the free space.

Localization

In order to test the quality of the proposed features, the localization approach has been tested for three cases:

- Vertical edges and corners only.
- Vertical edges, corners and empty spaces.
- Fused verticals/corners, verticals, corners and empty spaces.

For each of the ten test rooms, a fingerprint has been extracted after the robot has positioned itself in the center of the free space. This experiment has been repeated four times for each room. One fingerprint per room has then been included in a database as reference (map) for the localization approach. The other 30 fingerprints (3 per room) have been matched to the database for testing the localization. The results of the matching algorithm presented in section 2 are normalized to have a probability (between 0 and 1).

For a given observation (fingerprint), a match is successful if the best match with the database (highest probability) corresponds to the correct room. For the first experiment set we used only the vertical edges and the corners regrouped in a single fingerprint. In this case the percentage of successful matches is 64%. The second set of experiments introduced the empty space concept. The results improved with a percentage of successful matches of 73%. In the last experiment, the correspondences 'f' between the verticals and the corners has been added to the already tested features. The results have been further improved with a success rate of 77% (see Table 1).

Features set	Success rate
Verticals & corners	64%
Verticals, corners & empty spaces	73%
Fusion, verticals, corners & empty	77%

Table 1: The success rate depends on the chosen features set

Even if the presented method does not lead to a perfect success rate, it still delivers valuable information for false-matched rooms. This is an important characteristic of the presented method. When the room is successfully matched, the minimum energy algorithm gives a high probability: 0.83 in average and between 0.7 and 0.98. Even if it detects the correct room with the second or third highest probability, a localization approach, like for example a Partial Observable Markov Decision Process (POMDP) [5, 14] can use this information. In the experiments, the observations, which were not successfully matched, were in second or third place with a probability going from 0.5 to 0.7 (average 0.62). All these results are summarized in the following Table.

	MinProb	MaxProb	AvgProb
Successful match	0.70	0.98	0.83
Not successful match	0.50	0.70	0.62

Table 2: Probabilities for successful and unsuccessful matches

6. Conclusion and outlook

This paper presented the fingerprint concept as potential observation function within a topological framework for localization. The fingerprint structure with its circular sequence and the string-matching algorithm allow inserting any kind of features. This is shown in section 3, where four different kinds of feature from a laser scanner and an omnidirectional camera are presented. Using different features from multiple sensors allows improving the distinctiveness of the fingerprints as it as been proven in section 5.

The experiments show the robustness of the matching algorithm against occlusions and false detections, which often occur. By comparing 30 test fingerprints with the database composed of 10 rooms/fingerprints, the success rate (correct room with highest probability) is 77%. The remaining 23% also have high probability for the correct room (0.5 – 0.7). Even if the correct room has not the highest probability, this information can still be used for localization e.g. by employing a localization approach like a POMDP.

REFERENCES

[1] Abe, Y., Shikano, M.,Fukada, T., and Tanaka, Y., “Vision Based Navigation System for Autonomous Mobile Robot with Global Matching”, *IEEE International Conference on Robotics and Automation*, May 1999, pp. 1299-1304

[2] 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.

[3] 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.*

[4] Baeza-Yates, R., Navarro, G. ”Faster Approximate String Matching”, Department of Computer Science, University of Chile, Santiago.

[5] 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.

[6] Hough, P.V.C. “Method and Means for recognizing complex patterns”. *US Patent 3069654*, 1962.

[7] 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.*

[8] Nourbakhsh, I., J. Bodenage, et al. (1999). "An Affective Mobile Robot Educator with a Full-Time Job." *Artificial Intelligence 114*(1-2): 95-124.

[9] Kanade, T., Ohta, Y., “Stereo by Intra- and Inter-Scanline Search Using Dynamic Programming”, *IEEE Transactions on pattern analysis and machine intelligence*, Vol PALMZ No 3, March 1985

[10] 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.

[11] 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.

[12] Thrun, S., “Finding Landmarks for Mobile Robot Navigation”, *IEEE International Conf. on Robotics and Automation*, May 1998, pp. 958-963.

[13] 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.

[14] 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.*

[15] Ulrich, I. and I. Nourbakhsh (2000). “Appearance-Based Place Recognition for Topological Localization.” *IEEE International Conference on Robotics and Automation*, San Francisco, CA.