Robotica (2017) volume 35, pp. 254–270. © Cambridge University Press 2015
doi:10.1017/S0263574715000478

PCA-based method to fuse behaviors from place characterization for robot navigation

Alberto Poncela*

Dpto. Tecnología Electrónica, ETSI Telecomunicación
Universidad de Málaga, Campus de Teatinos, 29071, Málaga, Spain

(Accepted May 20, 2015. First published online: June 19, 2015)

SUMMARY

This paper presents a method to calculate the fusing rule among three reactive behaviors, Wall Following, Corridor Following and Door Crossing, from place characterization for robot navigation. The technique is supported by a local grid of the closest area to the robot, which is built from sonar readings. The contour of this grid is extracted, represented by its FFT and, finally, it is reduced to a short feature vector with a principal component analysis (PCA). This feature vector is used to decide the fusing rule among the three behaviors. The algorithm is very fast in terms of its time performance, being then valid to be used in robot navigation, since the robot would rapidly react to new situations. It has also been successfully tested in simulated and real environments, with a Pioneer robot equipped with eight frontal sonar sensors, both in manually driven tasks and autonomous navigation tasks, proving its feasibility and effectiveness.

KEYWORDS: Characterization; Behavior fusion; Robot navigation.

1. Introduction

Robot navigation is defined as the ability of a mobile robot to determine its own position and to reach a goal location in a safe way. This navigation might be performed through a behavior-based control. In fact, it has been proved that behaviors serve as building blocks for robotic actions.[1] According to,[2] a behavior is a control law that satisfies a set of constraints to achieve and maintain a particular goal. Behavior competition and behavior coordination are the two main approaches when dealing with robot behaviors. In the former, the system elicits one of the behaviors to accomplish the navigation task. This strategy is also known as behavior arbitration. Neural network,[3] fuzzy logic[4] and reinforcement learning[5] approaches have been reported. However, all of these techniques share the main drawback of competitive systems, how to decide which behavior should be selected at each time instant and the switching criteria among behaviors.

On the other hand, behavior coordination, also referred as behaviors fusion, deals with the simultaneous activation of several independent behaviors to obtain an emergent behavior to perform the desired navigation task.[6] As for behavior competition, many methods use the behaviors fusion paradigm, applying signal control merging,[7] fuzzy logic[8] or neural networks.[9] The main difficulty of these techniques is to find the cooperation rule to fuse all of the behaviors in an efficient way.[10]

This work is concerned with the decision on how to navigate with three behaviors: Wall Following, Corridor Following and Door Crossing, in a totally or partially unknown environment. Wall Following behavior moves the robot following the contour of its closest wall at a distance. Corridor Following behavior allows the robot to go across a corridor by its center. And, Door Crossing behavior allows the robot to cross a door by its middle. The difficulty is to decide which behavior to choose to reactively navigate at each time instant. If only one behavior is selected, it will not probably fit the environment configuration. However, a combination of the three behaviors would better suit any robot situation. Thus, behaviors fusion has been chosen.

* Corresponding author. E-mail: apg@dte.uma.es

https://doi.org/10.1017/S0263574715000478 Published online by Cambridge University Press

The strategy for behaviors fusion needs a mechanism to decide the fusing rule among the three behaviors. In this paper, a place characterization technique has been developed to fuse behaviors in an efficient way. Traditionally, place characterization has been treated as a visual-based problem, or reached either from laser data or sonar data. Visual approaches are usually considered as robust and reliable. Several authors follow this strategy, e.g.,.[11–13] Martínez's technique[11] is designed to recognize the current location of the robot to provide its heading. A supervised training stage is necessary. Furthermore, images must be captured carefully. This technique has two main drawbacks: (i) the training data must be large; and (ii) the training and test environments must be the same. Although Do's method[12] works from a small set of patterns, it shares the second disadvantage of Martínez's approach. The analysis of the work presented in[13] derives a similar conclusion. The training set is useful when recognizing places during the experiments, although both training and test environments cannot be different. Therefore, since in visual methods the training and the test environments must be the same and, therefore, they are not suitable for unknown environments. To solve this drawback, characterization from laser or sonar data has been proposed. Larbi[14] presents a system which can classify different situations during the robot navigation, but his method is very supervised and parameterized, having been tested only in simulated environments. Apart from the starting information, Martínez's approach[15] is similar to the proposal of this work. Places are categorized into rooms, hallways, doorways and corridors, even in unknown environments. However, the method needs 302 features to classify places and is not appropriate to fuse behaviors in a behavior-based navigation system, while this paper presents a method which works from a short feature vector (nine-dimensional) and, also, has been successfully applied to fuse behaviors in navigation. Other techniques[16] label environmental places semantically into rooms or corridors from a short feature vectors, as in this paper. Nevertheless, it has several drawbacks: (i) as reported in the study, the method does not work in unknown environments; (ii) the smallest processing time to categorize a place is 361 ms, which would potentially yield to an unsafety navigation; and (iii) behaviors would not be able to cooperate since the procedure strictly categorizes places into rooms or corridors. Sousa[17] also identifies rooms and corridors from a short 14-dimensional feature vector. Feature vectors consist of the result of a processing stage applied to a laser sensor scan. However, the method would only be valid for behavior competition, since places are strictly labeled as a room or a corridor. The number of categories is higher in other approaches,[18] although it must be pointed out that the method lacks tools to be applied in cooperative behavior-based navigation systems.

In this work, tests have been carried out with a Pioneer P2AT robot equipped with eight frontal sonar sensors and, thus, sonar sensors have been selected to acquire information from the environment. Sonar sensors provide distances to obstacles around the robot. However, they are prone to be affected by errors, specially angular inaccuracy and multiple echoes. Most sonar error correcting techniques rely on fusing sensor information in time and space, usually by means of evidence grids.[19] To overcome the sonar weakness and to achieve a better accuracy, techniques based on reprocessing incorrect sonar measurements[20] or Kalman filtering[21] can be helpful. Evidence grids are a multidimensional tessellation of the space into cells where each cell, with coordinates (x, y), presents a probability ranging from 0 to 1 of being occupied. Occupied areas present a probability 1 of being occupied, and free areas present a probability 0 of being occupied. Unexplored areas present a probability 0.5 of being occupied. Evidence grids are fast and easy to construct, efficiently combining any number of sensor readings. This work relies on evidence grids to achieve more stable sensor readings. However, grids involve a huge data volume and, hence, it is not immediate to compare them since they are affected by the robot heading and its relative position. Thus, grids are reduced, as an advantage, to a short feature vector by means of a PCA, used to characterize places to navigate fusing three behaviors, Wall Following, Corridor Following and Door Crossing. These three behaviors were chosen because they are usually used when exploring environments.[6,22,23]

Although the proposed technique is able to characterize a place into walls, corridors or doors in a behavior competition framework, in this work it has been used to cooperatively fuse the three behaviors since their combination will better suit any robot situation. The feature vector extraction is described in Section 2. This vector is used to decide the rule to fuse behaviors, as presented in Section 3. The importance of fusing behaviors is to provide the robot with the best motion command to navigate at each moment. Experiments and results obtained with this technique are shown in Section 4. Finally, conclusions are presented in Section 5.

https://doi.org/10.1017/S0263574715000478 Published online by Cambridge University Press

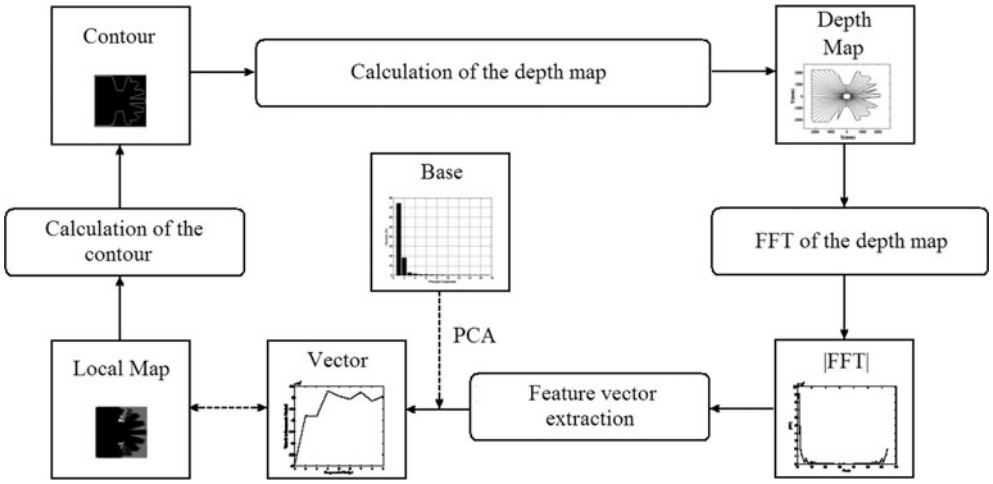


Fig. 1. Feature vector extraction algorithm.

2. Feature Vector Extraction

The process to extract the feature vector is depicted in Fig. 1. The local map is the input of the feature vector extraction algorithm, while the feature vector is its output. Next sections further describe the algorithm.

2.1. Calculation of the contour of the local map

The algorithm starts from a local grid built around the robot, representing its closest area, being the robot in the center. To define the closest area, 2 m around the robot have been considered. Since its size is 50×50 cm^2 and all the experiments were conducted with a cell size of 40 mm, the size of the local map is 113×113 cells.

The calculation of the contour of the local map relies on the extraction of the Region Free of Obstacles (RFO) in the local map. The RFO involves a huge data volume. Thus, it is represented by its contour $\overrightarrow{C}$, through the following phases:

(1) *Thresholding*. Cells whose occupancy value is below a threshold U_{occ} are considered free space. U_{occ} has been empirically set to 0.4. The rest will be treated as occupied or non-explored. The thresholded local map is stored in polar coordinates to easily look for the closest non-free cell $O\phi$ to the robot in each direction ϕ.
(2) *Filtering*. As the thresholded grid is noisy, a median filter is used to partially remove it from the representation.
(3) *Obtaining the RFO*. The previous stage yields a local map with a set of non-connected regions of free space. This set of regions is analyzed to discriminate the RFO around the robot.
(4) *Calculation of the contour*. Finally, the RFO is processed to obtain the contour $\overrightarrow{C}$ of the local map. The contour $\overrightarrow{C}$ consists of a K-dimensional vector of points. Each point C_k is determined by its (x, y) coordinate in the image of the contour. Thus, the contour $\overrightarrow{C}$ will be

$$\overrightarrow{C} = [C_1, C_2, \ldots, C_K] = [(x_1, y_1), (x_2, y_2), \ldots, (x_K, y_K)]. \quad (1)$$

Figures 2(a), (e) and (i) show three local maps acquired near a wall, in a corridor and for a door, respectively. Obstacles are represented in white, free space in black and non-explored areas in gray. In this work, both simulated and real tests have been carried out with a local map of 113×113 cells and a cell size of 40 mm. The contours of the local maps in Figs. 2(a), (e) and (i) are represented in Figs. 2(b), (f) and (j), yielding 338, 423 and 614 points, respectively, only a 2.64%, a 3.31% and a 4.80% of the data volume of the initial local maps.

https://doi.org/10.1017/S0263574715000478 Published online by Cambridge University Press

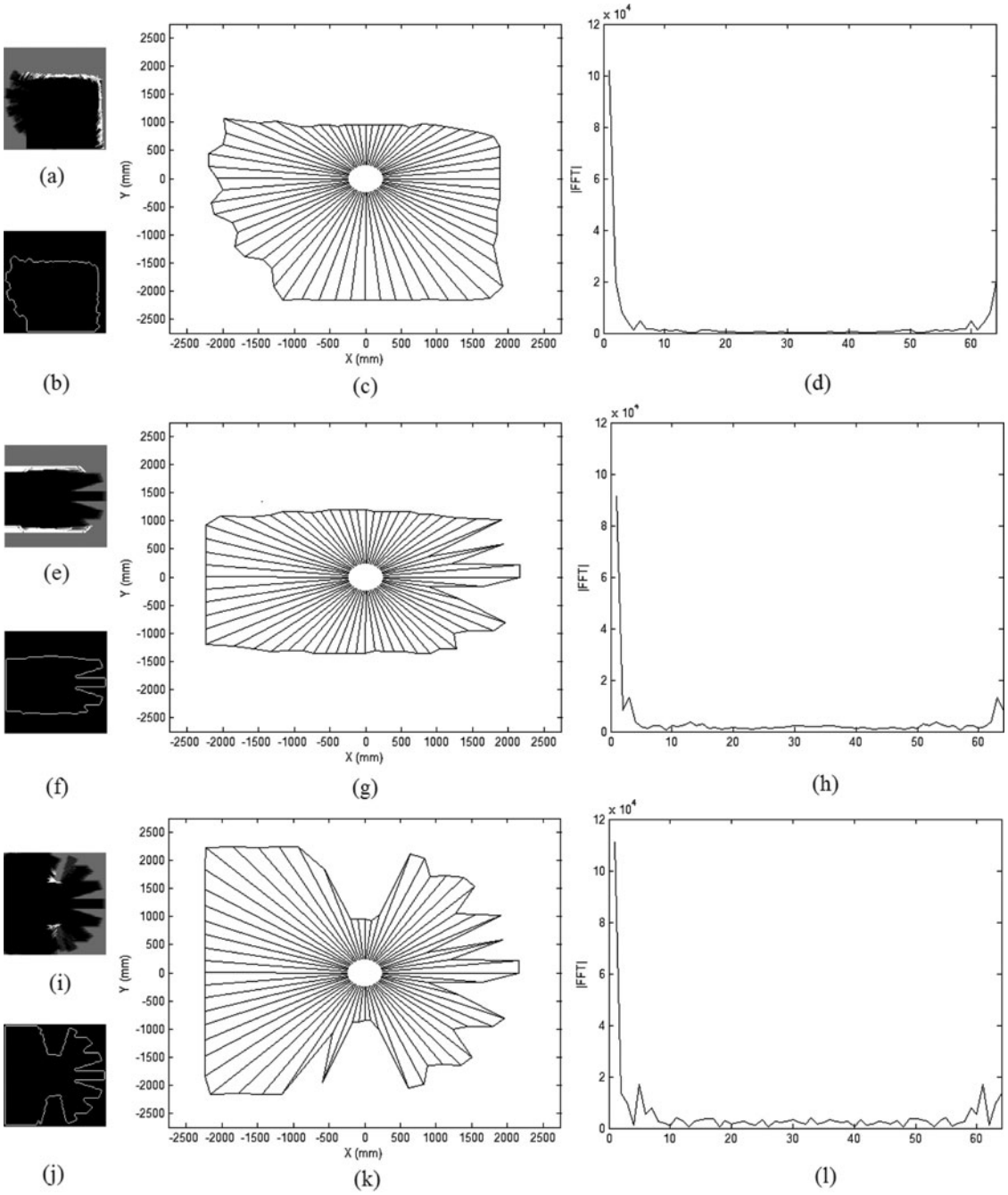


Fig. 2. Wall: (a) local map; (b) contour of the RFO of the local map; (c) depth map; (d) module of the FFT of the depth map; Corridor: (e) local map; (r) contour of the RFO of the local map; (g) depth map; (h) module of the FFT of the depth map; Door: (i) local map; (j) contour of the RFO of the local map; (k) depth map; (l) module of the FFT of the depth map.

2.2. Depth map calculation

Vector $\overrightarrow{C}$ has a variable length, depending on the local map. Thus, it is not suitable for PCA processing and, hence, it is reduced in this stage to a N-dimensional vector, the depth map. In this work $N = 64$, since it is a power of two and the next step, FFT, can then be efficiently applied.

A depth map, $\overrightarrow{DM} = [DM_1, DM_2, \ldots, DM_N]$, is a N-dimensional vector which models the distance to obstacles that a ring of N equally spaced sensors would measure in the robot position. Let $\overrightarrow{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_N]$ be the orientation of each of the N equally spaced sensors, and $\triangle\alpha = \frac{360^\circ}{N}$

https://doi.org/10.1017/S0263574715000478 Published online by Cambridge University Press

be the angle difference between two consecutive sensors. Contour $\overrightarrow{C}$ is processed to calculate a N-dimensional vector $\overrightarrow{C}^N = [C_1^N, C_2^N, \ldots, C_N^N] = [(x_1^N, y_1^N), (x_2^N, y_2^N), \ldots, (x_N^N, y_N^N)]$. $\overrightarrow{C}^N$ contains those points belonging to the contour $\overrightarrow{C}$ which orientations are closer to angles in vector $\overrightarrow{\alpha}$. If (c_x, c_y) is the center of the local map, points are obtained with the following expression:

$$C_i^N = C_k / |\arctan\frac{y_k - c_y}{x_k - c_x} - \alpha_i| = \underset{1\le m\le K}{min} |\arctan\frac{y_m - c_y}{x_m - c_x} - \alpha_i|. \tag{2}$$

Then, the distance DM_i of sensor i with orientation α_i is

$$DM_i = \left[\sqrt{(x_i^N - c_x)^2 + (y_i^N - c_y)^2}\right] \times Size, \tag{3}$$

where $Size = 40$ mm is the cell size used in all the experiments. Figures 2(c), (g) and (k) show the depths map obtained from contours in Figs. 2(b), (f) and (j), respectively. It must be pointed out that a depth map implies a reduction of the data volume stored in the local map by 99.5%.

2.3. FFT of the depth map

Depth maps are represented by the module of its fast fourier transform (FFT), because it is invariant to rotations. Let $\overrightarrow{FFT} = [FFT_1, FFT_2, \ldots, FFT_N]$ be the vector with the FFT of the depth map $\overrightarrow{DM}$, and $|\overrightarrow{FFT}| = [|FFT_1|, |FFT_2|, \ldots, |FFT_N|]$ be the vector with the module of the FFT. Each point of the vector $\overrightarrow{FFT}$, FFT_n, is given by

$$FFT_n = \sum_{k=1}^{N} DM_k e^{-2\pi j(k-1)(n-1)/N}. \tag{4}$$

Figures 2(d), (h) and (l) show the module of the FFT of the depth maps in Figs. 2(c), (g) and (k), respectively. The module of the FFT yields a large number of zero or almost zero values in the three cases (wall, corridor and door), being the relevant information at low frequencies.

2.4. Short feature vector

Considering that most of the information in the $|FFT|$ is at low frequencies, the size of the vector to characterize a place can be further reduced. To that purpose, a PCA is applied. It is a technique to reduce the dimensionality of a data set of correlated variables while retaining most of their original variations. PCA has been previously used to reduce data from visual sensors,[24,25] laser sensors[26] or sonar sensors.[27,28]

The set of all possible $|FFT|$s of depth maps calculated from local maps presenting a length of N can be defined in a space of dimension N. Most of the values in vector $|\overrightarrow{FFT}|$ are almost zero. Thus, all possible $|\overrightarrow{FFT}|$ conform a P-dimensional vectorial subspace, being $P < N$. Let $\{|\overrightarrow{FFT}|_m\}_{m=1}^M$ be a set of M $|FFT|$s of the depth map of the contour of the RFO of M local maps. Considering that all $|FFT|$s belong to a P-dimensional subspace, a base $\{\overrightarrow{\Phi}_k\}_{k=1}^P$ is needed:

$$|\overrightarrow{FFT}|_i \cong \sum_{k=1}^{P} \beta_{ik} \overrightarrow{\Phi}_k, \tag{5}$$

where $\{\beta_{ik}\}_{k=1}^P$ is the feature vector of principal components of vector $|\overrightarrow{FFT}|_i$, obtained when $|\overrightarrow{FFT}|_i$ is projected onto the base $\{\overrightarrow{\Phi}_k\}_{k=1}^P$. The mean square error will be minimum if the base is optimal, that is, when it is extracted from the set of M eigenvectors of the covariance matrix of $\{|\overrightarrow{FFT}|_m\}_{m=1}^M$:

$$C_{FFT} = \frac{1}{M} \sum_{i=1}^{M} |\overrightarrow{FFT}|_i \cdot |\overrightarrow{FFT}|_i^T. \tag{6}$$

https://doi.org/10.1017/S0263574715000478 Published online by Cambridge University Press

C_{FFT} has M eigenvalues. Eigenvalues sorted from highest to lowest, $\lambda_1 > \lambda_2 > \cdots > \lambda_M$, have their corresponding eigenvector $\overrightarrow{\Phi}_1, \overrightarrow{\Phi}_2, \ldots, \overrightarrow{\Phi}_M$. The base $\{\overrightarrow{\Phi}_k\}_{k=1}^{P}$ consist of the first P eigenvectors. The projection onto the base of each vector $\overrightarrow{|FFT|}_i$ of the initial set of vectors $\{\overrightarrow{|FFT|}_m\}_{m=1}^{M}$ yields its principal components vector, $\overrightarrow{\beta}_i$, which is its feature vector, $\overrightarrow{FV}_i$

$$\overrightarrow{FV}_i = \overrightarrow{\beta}_i = [\beta_{i1}, \beta_{i2}, \ldots, \beta_{iP}]. \tag{7}$$

The feature vector $\overrightarrow{FV}_j = [FV_{j1}, FV_{j2}, \ldots, FV_{jP}]$ of principal components of any vector $\overrightarrow{|FFT|}_j$ not included in the initial set of vectors $\{\overrightarrow{|FFT|}_m\}_{m=1}^{M}$ is obtained through its projection onto the base. FV_{jk} is calculated projecting vector $\overrightarrow{|FFT|}_j$ onto vector $\overrightarrow{\Phi}_k$ of the base $\{\overrightarrow{\Phi}_k\}_{k=1}^{P}$

$$FV_{jk} = \overrightarrow{|FFT|}_j \cdot \overrightarrow{\Phi}_k = \sum_{m=1}^{N} |FFT|_{jm} \Phi_{km}. \tag{8}$$

The base was calculated only once by PCA and off-line. It has been heuristically tested that it is not necessary to use a high number of local maps to achieve a suitable base. The base was obtained from 17 randomly simulated local maps, including walls, corridors, doors and maps do not clearly belong to any of these configurations. This base was used in all the experiments. PCA and an off-line base calculation has been previously used,[29] using only a few samples for this off-line stage. In this work, tests were carried out in real and simulated environments to prove that the base is valid for environments different from those where it was calculated. Even when the base was obtained from simulated maps, it has been successfully used in real scenarios. To select the dimension P of the vectorial subspace of the base $\{\overrightarrow{\Phi}_k\}_{k=1}^{P}$, the amount of information explained by the principal components is analyzed, as other methods do.[30] The percentage of variance explained by the first m principal components is:

$$\sum_{k=1}^{m} Var(FV_k) = 100 \frac{\sum_{k=1}^{m} \lambda_k}{\sum_{k=1}^{P} \lambda_k}. \tag{9}$$

Figure 3 shows the percentage of information explained by the first m principal components. It can be appreciated that the first component explains most of the information (75%), while the percentage of variance explained by the first nine components is a 99.5%. Thus, all our experiments have been carried out with a nine-dimensional base $\{\overrightarrow{\Phi}_k\}_{k=1}^{9}$. In the system, any local map is represented by its nine-dimensional feature vector through the projection of the $|FFT|$ of the depth map of the contour of the RFO onto the base, reducing the initial size of the grid to a short efficient vector. In fact, the size of the nine-dimensional feature vector is only a 0.07% of the initial local map.

3. Behavior Fusion

A feature vector must be analyzed to characterize a place. This analysis yields a place characterization in wall, corridor or door category. The classification of places in wall, corridor or door would allow to select the best behavior for robot navigation, Wall Following, Corridor Following or Door Crossing. Although the proposed feature vector can be treated as a behaviors competition framework, it is also suitable to fuse behaviors since a combination of the three behaviors would better suit any robot situation. Thus, in this work, after a feature vector is analyzed, the weighting factors which should be applied to efficiently combine the three behaviors are calculated. Next sections describe, firstly, the feature vector matching process and, then, the technique to obtain the three weighting factors.

https://doi.org/10.1017/S0263574715000478 Published online by Cambridge University Press

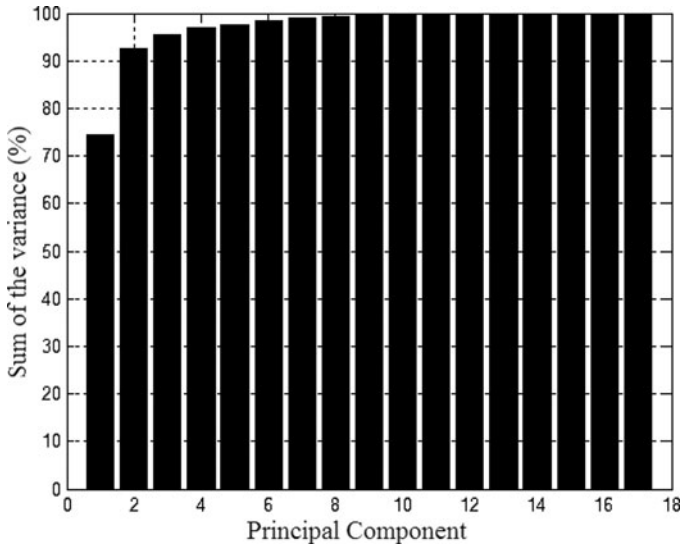


Fig. 3. Percentage of variance explained by the first m principal components.

3.1. Feature vector matching

Feature vector matching is based on the comparison between the feature vector of a local map and a set of known and predefined patterns. Patterns are the feature vectors extracted from simulated local maps. Nine patterns have been experimentally chosen in the system, three for each of the categories, wall, corridor and door. They are calculated only once and off-line. The starting point to obtain the patterns are the nine local maps presented in Fig. 4(a). Wall patterns were selected to include three different situations, while corridor patterns consider three wideness and door patterns are concerned with three typical scenarios near a door. The number of patterns is significantly low, while other methods need a large database to characterize places.[13] Patterns have been obtained from purposefully captured simulated local maps to prove that the method can be easily applied to different robot platforms, even in real scenarios. Furthermore, it must be pointed out that local maps used to calculate the patterns are different from those used to obtain the base. Thus, the base calculation is independent from the pattern selection and, hence, a new set of patterns could be selected without recalculating the base, making the method more flexible and adaptable.

The matching process consists of comparing a feature vector with the nine predefined patterns. The Tanimoto distance[31] has been chosen because local maps at similar locations yield parallel vectors. The Tanimoto distance favors the similarity and correlation between feature vectors, instead of their Euclidean distance. Let $\overrightarrow{FV}_i = [FV_{i1}, FV_{i2}, \ldots, FV_{iP}]$ and $\overrightarrow{FV}_j = [FV_{j1}, FV_{j2}, \ldots, FV_{jP}]$ be two feature vectors. The Tanimoto distance between $\overrightarrow{FV}_i$ and $\overrightarrow{FV}_j$ is defined as

$$D(\overrightarrow{FV}_i, \overrightarrow{FV}_j) = 1 - \frac{\overrightarrow{FV}_i^T \cdot \overrightarrow{FV}_j}{\left|\overrightarrow{FV}_i\right|^2 + \left|\overrightarrow{FV}_j\right|^2 - \overrightarrow{FV}_i^T \cdot \overrightarrow{FV}_j}. \tag{10}$$

To prove the suitability of the method to discriminate places, Fig. 4(b) shows nine local maps captured in real environments, including walls (Maps 1–3), corridors (Maps 4–6) and doors (Maps 7–9). Their corresponding feature vectors have been compared with the nine patterns to characterize the places they represent. This comparison is depicted in Fig. 5. For clearer representation, the feature vector of each local map is only compared with the patterns of the category it is supposed to belong to. Figures 5(a)–(c) show that Maps 1, 2 and 3 yield feature vectors similar to the three wall patterns (Patterns 1, 2 and 3). The same conclusion can be extracted when comparing the feature vectors of Maps 4, 5 and 6 with the corridor patterns (Figs. 5(d)–(f)). And, for door configurations (Maps 7, 8 and 9), the comparison is established with Patterns 7, 8 and 9 (Figs. 5(g)–(i)).

Although the qualitative comparison between a feature vector and the patterns helps in characterizing a place, the decision is based on the Tanimoto distance between a feature vector

https://doi.org/10.1017/S0263574715000478 Published online by Cambridge University Press

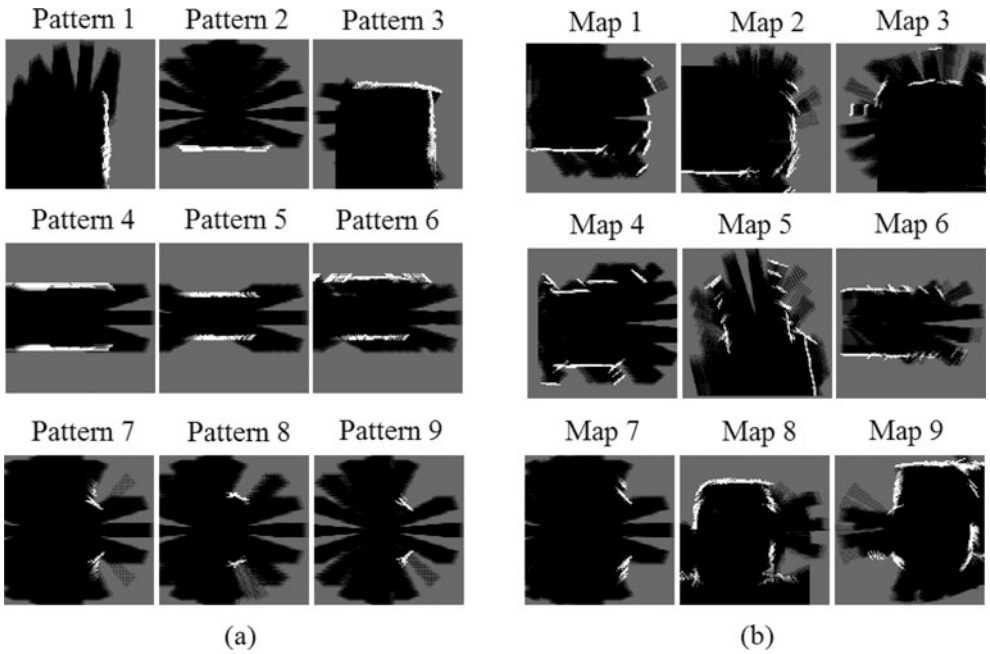


Fig. 4. (a) Local maps of the patterns. Patterns 1–3: wall; Patterns 4–6: corridor; Patterns 7–9: door. (b) Test local maps. Maps 1–3: wall; Maps 4–6: corridor; Maps 7–9: door.

Table I. Tanimoto distances from the feature vectors of the local maps in Fig. 4(b) (rows) to the 9 patterns (columns).

	Pat. 1	Pat. 2	Pat. 3	Pat. 4	Pat. 5	Pat. 6	Pat. 7	Pat. 8	Pat. 9
a	0.18	0.15	**0.14**	0.67	0.71	0.36	0.32	0.37	0.47
b	0.24	0.07	**0.06**	0.80	0.83	0.52	0.30	0.47	0.40
c	0.32	**0.10**	0.11	0.86	0.89	0.62	0.37	0.40	0.48
d	0.78	0.79	0.82	**0.25**	0.36	0.36	0.79	0.61	0.82
e	0.50	0.60	0.68	0.16	0.19	**0.08**	0.79	0.48	0.85
f	0.75	0.84	0.90	**0.12**	0.24	0.28	0.90	0.74	0.92
f	0.42	0.38	0.33	0.92	0.97	0.71	**0.09**	0.13	0.11
h	0.45	0.39	0.40	0.54	0.61	0.39	**0.27**	0.37	0.33
i	0.65	0.48	0.40	0.84	0.87	0.68	**0.16**	0.39	0.19

and the patterns. Table I presents the Tanimoto distance between the feature vectors of the real local maps in Fig. 4(b) and the patterns. The minimum distance for each local map is shown in bold. These minimum distances lead to the category each local map belongs to. Maps 1–3 would be categorized as walls, Maps 4–6 as corridors and Maps 7–9 as doors. It must be noted that minimum distances range from 0.06 to 0.27. These distances are low, making the matching process very accurate. For example, the Tanimoto distance between the feature vector of the wall represented in Map 2 of Fig. 4(b) and the patterns ranges from 0.06 to 0.83. For the same map, if only the wall category is considered, the Tanimoto distance ranges from 0.06 to 0.24. Thus, the minimum value (Tanimoto distance) is 0.06 and, hence, Map 2 may be categorized as a wall.

3.2. Weighting factors

The feature vector matching process is able to characterize a place into wall, corridor or door category. This characterization would allow Wall Following, Corridor Following and Door Crossing behaviors to compete in robot navigation. However, in this work, the proposed feature vector is used to decide the fusing rule among the three behaviors to fit the environment configuration at each time instant.

Let $\overrightarrow{D} = [d_1, d_2, d_3, d_4, d_5, d_6, d_7, d_8, d_9]$ be the vector with the Tanimoto distances from the feature vector $\overrightarrow{FV}$ of a local map to the nine patterns of the system. Let also $\overrightarrow{W}_C =$

https://doi.org/10.1017/S0263574715000478 Published online by Cambridge University Press

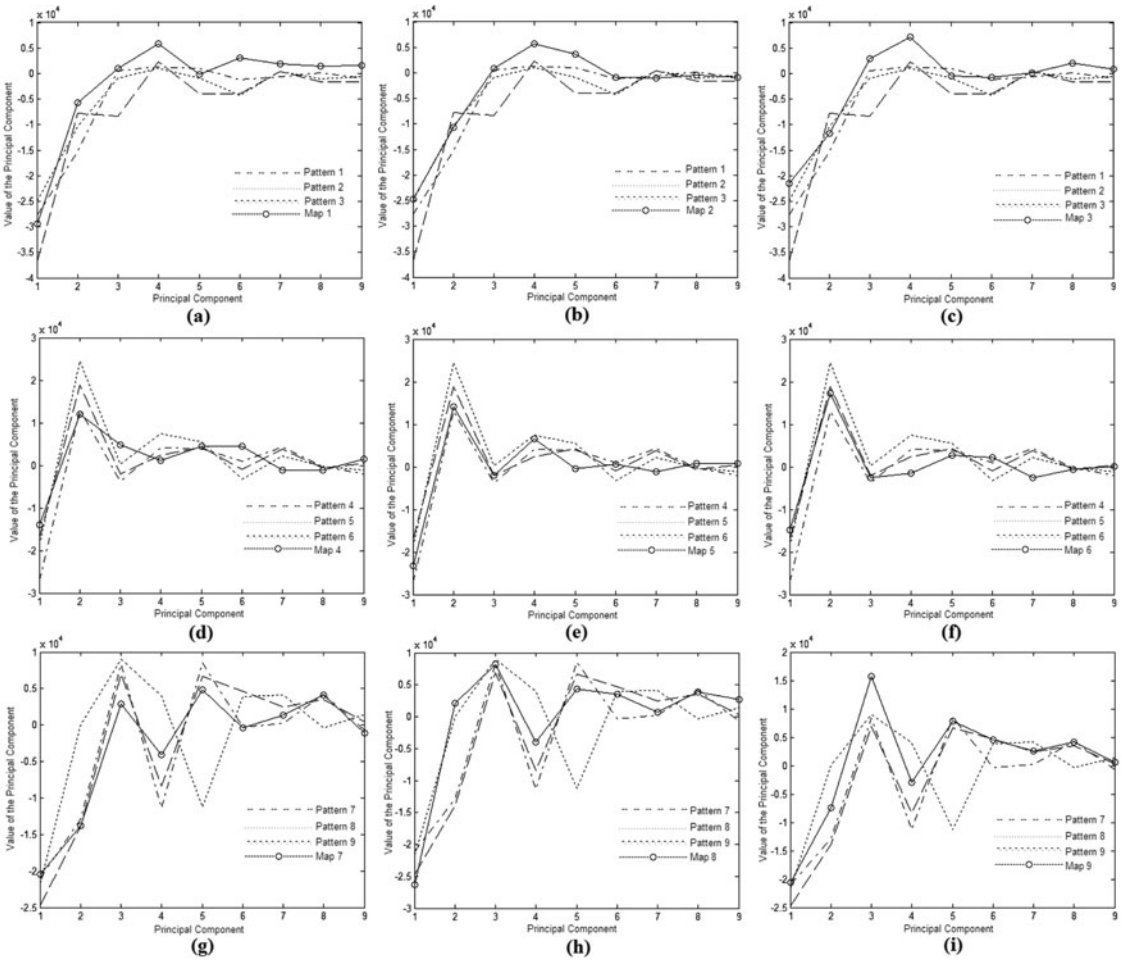


Fig. 5. Comparison of the feature vectors of the local maps in Fig. 4(b) with the patterns: (a) Map 1; (b) Map 2; (c) Map 3; (d) Map 4; (e) Map 5; (f) Map 6; (g) Map 7; (h) Map 8; (i) Map 9.

$[w_{wall}, w_{corridor}, w_{door}]$ be the vector with the weighting factors to fuse the three behaviors in a single robot motion command, such that $0 \leq w_{wall} \leq 1$, $0 \leq w_{corridor} \leq 1$, $0 \leq w_{door} \leq 1$ and $w_{wall} + w_{corridor} + w_{door} = 1.0$.

$\overrightarrow{W}_C$ is derived from $\overrightarrow{D}$. Let $d^{min}_{wall} = min\{d_1, d_2, d_3\}$, $d^{min}_{corridor} = min\{d_4, d_5, d_6\}$ and $d^{min}_{door} = min\{d_7, d_8, d_9\}$ be the minimum distance to the patterns of each category, wall (Patterns 1–3), corridor (Patterns 4–6) and door (Patterns 7–9), respectively. Let also d^{min} be the minimum of these three values, $d^{min} = min\{d^{min}_{wall}, d^{min}_{corridor}, d^{min}_{door}\}$.

The mean values from the distances to the patterns of each category, wall, corridor and door is, respectively, $\overline{d_{wall}}$, $\overline{d_{corridor}}$ and $\overline{d_{door}}$. Then, if $d^{min} = d^{min}_{wall}$, the vector with the weighting factors, $\overrightarrow{W}_C = [w_{C1}, w_{C2}, w_{C3}]$, will be

$$\overrightarrow{W}_C = [1.0, d^{min}_{wall}/\overline{d_{corridor}}, d^{min}_{wall}/\overline{d_{door}}]. \tag{11}$$

If $d^{min} = d^{min}_{corridor}$, it will be

$$\overrightarrow{W}_C = [d^{min}_{corridor}/\overline{d_{wall}}, 1.0, d^{min}_{corridor}/\overline{d_{door}}]. \tag{12}$$

And if $d^{min} = d^{min}_{door}$, it will be given by

$$\overrightarrow{W}_C = [d^{min}_{door}/\overline{d_{wall}}, d^{min}_{door}/\overline{d_{corridor}}, 1.0]. \tag{13}$$

https://doi.org/10.1017/S0263574715000478 Published online by Cambridge University Press



Fig. 6. Pioneer P2AT in a test environment.

To comply with $w_{C1} + w_{C2} + w_{C3} = 1.0$, $\Sigma = \sum_{i=1}^{3} w_{Ci}$ is the normalization factor. The weighting factors the system would apply to linearly fuse the three behaviors in one motion command are

$$\overrightarrow{W}_C = [w_{wall}, w_{corridor}, w_{door}] = \left[\frac{w_{C1}}{\Sigma}, \frac{w_{C2}}{\Sigma}, \frac{w_{C3}}{\Sigma}\right]. \tag{14}$$

4. Results and Discussion

The proposed technique has been tested for several different simulated and real environments, with a Pioneer P2AT robot equipped with eight frontal Polaroid sonar sensors (Fig. 6). The experiments were carried out for a local map yielding 113×113 cells and a cell size of 40 mm. The local map is built incrementally while the robot navigates. As aforementioned, depth maps and the module of their corresponding FFTs are 64-dimensional vectors, whereas the base to calculate the feature vectors is nine-dimensional. All the experiments were performed on a PC under Linux operating system (Ubuntu 12.04), an Intel Core i5 3.2 GHz with 4 GB DDR memory.

In order to evaluate the results, a graphical representation has been developed, the Behaviors Fusion Representation (BFR). Each category is assigned a RGB color, red for wall, green for corridor and blue for door, that is, each weighting factor is assigned to a RGB color channel, such that $[R, G, B] = [255 \cdot w_{wall}, 255 \cdot w_{corridor}, 255 \cdot w_{door}]$. The BFR consists of the robot trajectory, in RGB color, superimposed to an evidence grid that the robot builds while it navigates. In an evidence, grid obstacles are represented in white, free space in black and non-explored areas in gray.

In next sections, the results of the proposed method are presented. Prior to the experiments in navigation, the time performance of the algorithm is analyzed. Then, tests when the robot navigates are presented in two cases, for a manually driven agent and, finally, for a fully autonomous robot.

4.1. Time performance

Table II shows the average time to calculate the contour of the RFO of a local map, whereas Table III presents the average time of the whole algorithm to obtain the weighting factors to apply to fuse the three behaviors, Wall Following, Corridor Following and Door Crossing. Three local maps have been considered, belonging to the wall category (Fig. 2(a)), corridor category (Fig. 2(e)) and door category (Fig. 2(i)). Errors have been calculated with a confidence interval of a 95%.

The first conclusion derived from Tables II and III is that the time consumption is independent of the environment configuration. It must also be pointed out that the calculation of the contour is the most time-consuming process, around 1.7 ms. The remaining steps of the algorithm use a very reduced data volume. Thus, the time consumption for those phases is significantly lower.

https://doi.org/10.1017/S0263574715000478 Published online by Cambridge University Press

Table II. Time performance: calculation of the contour of the local map.

Local map	Fig. 2(a)	Fig. 2(e)	Fig. 2(i)
Calculation of the contour of the local map			
1. Thresholding	0.760 ms ± 5.0 µs	0.755 ms ± 5.0 µs	0.754 ms ± 3.0 µs
2. Filtering	0.421 ms ± 2.4 µs	0.378 ms ± 4 µs	0.468 ms ± 1.9 µs
3. Obtaining the *RFO*	0.248 ms ± 2.1 µs	0.261 ms ± 3 µs	0.247 ms ± 0.7 µs
4. Calculation of the contour	0.254 ms ± 1.6 µs	0.214 ms ± 1.8 µs	0.296 ms ± 0.8 µs
Total contour	1.683 ms ± 10 µs	1.608 ms ± 13 µs	1.766 ms ± 6 µs

Table III. Time performance: calculation of the weighting factors to fuse behaviors.

Local map	Fig. 2(a)	Fig. 2(e)	Fig. 2(i)
Total contour	1.683 ms ± 10 µs	1.608 ms ± 13 µs	1.766 ms ± 6 µs
Depth map calculation	0.081 ms ± 0.4 µs	0.100 ms ± 0.6 µs	0.149 ms ± 0.3 µs
$\|FFT\|$ of the depth map	3.3 µs ± 0.1 µs	3.3 µs ± 0.1 µs	3.3 µs ± 0.04 µs
Feature vector	0.620 µs ± 0.03 µs	0.630 µs ± 0.03 µs	0.630 µs ± 0.03 µs
Weighting factors	0.425 µs ± 0.03 µs	0.420 µs ± 0.03 µs	0.470 µs ± 0.03 µs
Total	1.770 ms ± 11 µs	1.713 ms ± 14 µs	1.920 ms ± 6 µs

Since the contour $\overrightarrow{C}$ has a variable length, the time to obtain the depth map is variable, as well. Thereby, the contours of the local map in Figs. 2(a), (e) and (i) yield 338, 423 and 614 points, respectively. Therefore, the time consumption shown in Table III ranges from 0.081 ms (338 points) to 0.149 ms (614 points). After the depth map calculation has finished, the method works with 64-dimensional vectors. Thus, the time consumption is similar in the three cases, as presented in Table III.

The last three stages contribute with a very reduced processing time. First, the calculation of the $|FFT|$ of the depth map is very fast, because of the 64-dimensional vectors involved. In the three cases, the technique executes this step in only 3.3 µs. The projection of these $|FFT|$ onto the base is very fast, as well, needing only 0.630 µs. This time does not include the base calculation, since it is obtained only once and off-line. Finally, the weighting factors stage needs less than 0.48 µs. This final phase considers both the calculation of the Tanimoto distance from the feature vector to the patterns and obtaining the weights to fuse the three motion commands of the behaviors.

The total time of the algorithm is below 2 ms. If the robot navigates at a maximum linear velocity of 100 mm/s, the platform walks only 0.2 mm before a new calculation of the weighting factors. Even if the velocity was much higher, e.g., 1 m/s, the robot would only walk 2 mm before a new calculation. Hence, as the method is as fast, the reaction to new situations in very high. Besides, the robot could adapt to the environment configuration, independently of its shape. Thus, the proposed method allows its use in robot navigation, because the robot could perfectly react to new and unexpected situations in a rapid way.

4.2. Guided navigation

The robot was manually driven with a keyboard at a maximum angular velocity of 15°/s and a maximum linear velocity of 100 mm/s. The test environments are purposefully different from those used to obtain the base and from those where the local maps to calculate the nine patterns were captured.

A first simulated environment is shown in Fig. 7(a). It consists of a square room with an obstacle in the center. The robot was guided from the south to the east, then to the north and, finally, to the west, as it is depicted in the robot trajectory of the BFR in Fig. 7(b). When the robot moves between the obstacle and a wall, the environment is perceived as a corridor (green color). When the robot is approaching a front wall and, hence, leaving the obstacle, the robot determines that it is navigating through a door (blue color). And when it is turning, walls are detected (red color). The three weighting

https://doi.org/10.1017/S0263574715000478 Published online by Cambridge University Press

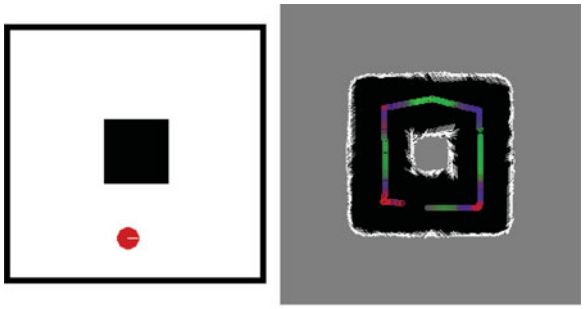


Fig. 7. Simulated environment: (a) configuration; (b) BFR.

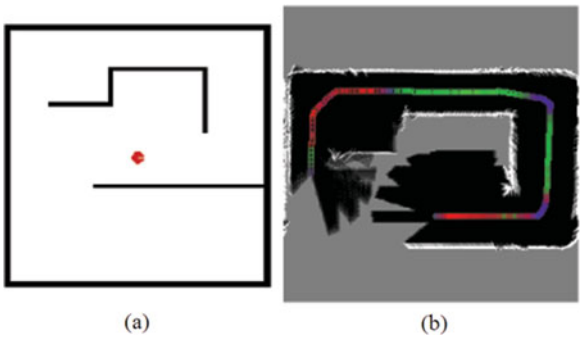


Fig. 8. Simulated environment: (a) configuration; (b) BFR.

factors, w_{wall}, $w_{corridor}$ and w_{door} have an influence at each point of the robot trajectory, yielding a percentage of participation for the three behaviors if they were merged. Thereby, Corridor Following behavior would have the main contribution when the robot is navigating between the obstacle and a wall. However, this configuration could also be considered as a wall and, somehow, as a door. This happens, e.g., when the agent moves between the obstacle and the east wall. In this situation, the method obtains the vector $\overrightarrow{W}_C$ with the corresponding weighting factors, $\overrightarrow{W}_C = [0.67, 0.17, 0.16]$, being the Corridor Following behavior the one which would contribute most (67%) to the robot motion command.

Another simulated environment is shown in Fig. 8(a), while its BFR is represented in Fig. 8(b). The guiding process started with the robot in the center heading east. It was guided towards the east, to the north when the east wall was faced, then to the west and, at the end, to the south. Although the configuration differs from the first simulated environment (Fig. 7(a)), the BFR allows to extract similar conclusions. When the robot is between two obstacles, the environment is as a corridor (green color), that is, through the right and the north corridors. Furthermore, when the robot is moving between two obstacles and a wall is detected in the front, the environment configuration is perceived as a door (blue color). Finally, when there is enough free space on the left with an obstacle on the right, the robot is near a wall. This situation appears at the beginning and when the robot turns from the north to the west. However, as aforementioned, behaviors do not compete but cooperate. Therefore, their weighting factors would provide the final robot motion command at each time instant. In this sense, when the robot navigates through the north corridor, vector $\overrightarrow{W}_C = [0.11, 0.77, 0.12]$ determines that the Wall Following, Corridor Following and Door Crossing behaviors would contribute with a 11%, a 77% and a 12% to the motion command, respectively. Door Crossing behavior contributes with the highest weighting factor when the robot turns to the north and to the west (blue color). In both turns, w_{door} is above 0.6 (60%).

A second set of experiments was conducted in the real world. The robot was firstly guided in simple scenarios. Figure 9(a) presents a real environment, where the robot was guided parallel to the wall. The wall is in a 4.4×4.4 m^2 room. The BFR is shown in Fig. 9(b). The robot clearly detects the

https://doi.org/10.1017/S0263574715000478 Published online by Cambridge University Press

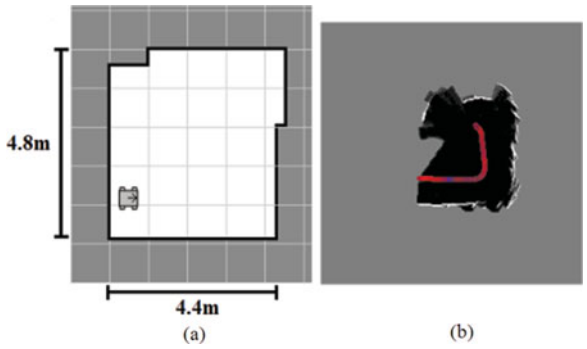


Fig. 9. Real environment: (a) configuration; (b) BFR.

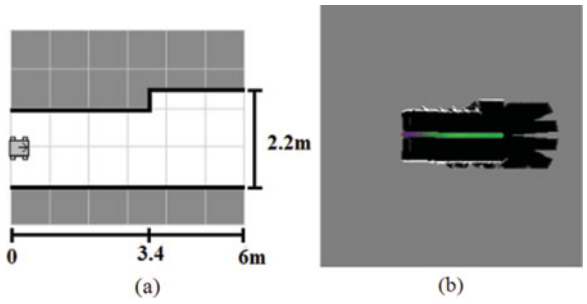


Fig. 10. Real environment: (a) configuration; (b) BFR.

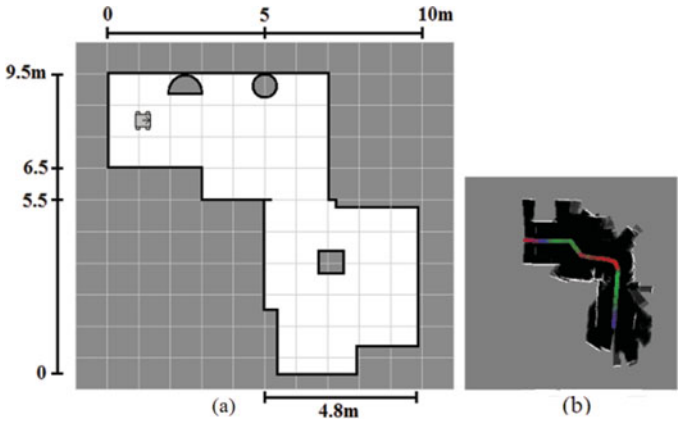


Fig. 11. Real environment: (a) configuration; (b) BFR.

wall configuration in almost the whole trajectory. The wall category ranges from a 63% to a 88%, while for the corridor it is between a 4% and a 15%. For the door one, it is between a 8% and a 22%.

Another simple environment is presented in Fig. 10(a). The robot was guided along the corridor from west to east.The corridor is 5 m long and 1.8 m of 2.2 m wide. In Fig. 10(b), the BFR after guiding the robot is shown. At the beginning, the main category is incorrect, since the local map is not sufficiently conformed. While the robot moves, the local map is incrementally updated. Hence, the corridor is perfectly detected, ranging $w_{corridor}$ from 0.7 (70%) to 0.85 (85%). Thus, the Corridor Following behavior would have a significantly higher contribution than those for the other behaviors, yielding an intense green color in the BFR (Fig. 10(b)).

The configuration of a third real environment is depicted in Fig. 11(a). It consists of a corridor and a room with a centred small obstacle (60×60 cm^2). The obtained BFR is presented in Fig. 11(b). The robot was guided along the corridor from west to east. Then, the agent entered the room through the door and continued by the west side of the obstacle.

https://doi.org/10.1017/S0263574715000478 Published online by Cambridge University Press

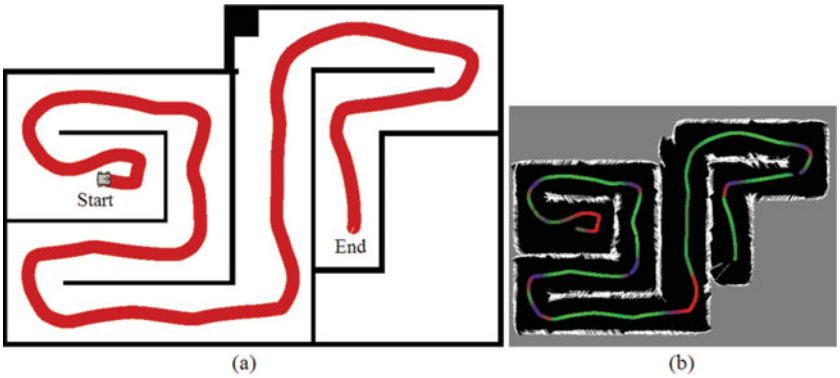


Fig. 12. Simulated environment: (a) configuration; (b) BFR.

Initially, the local map is not totally built and, thus, the wall category is the main (red color), followed by the door one (blue color). When the corridor appears, our technique detects this situation (green color). Next, the robot moves parallel to the right wall (red color). After that, the robot comes into the room. It sees the environment as a corridor, since the map presents a wall both on its left and on its right. Finally, when the centerd obstacle appears, it seems to be a door. When the robot moves along the north corridor and it turns right to follow the wall, w_{wall} ranges from 0.55 (55%) to 0.78 (78%), whereas when the obstacle in the center of the room appears, the robot thinks that it is moving through a door. This situation yields a door category percentage from a 45% to a 65%.

4.3. Autonomous navigation

In these tests, the robot navigated autonomously to perform a behavior-based exploration task. At each point of the robot trajectory, the system combined the motion command of the three behaviors, Wall Following, Corridor Following and Door Crossing. These motion commands were calculated as presented in.[32] Each behavior was assigned a weight, being determined by the weighting factors the proposed method calculates, w_{wall}, $w_{corridor}$ and w_{door}. The cooperation among the three behaviors is achieved through the linear combination of their motion commands. The final motion command to apply at each time instant is defined by:

$$v_t = w_{wall} \cdot v_{WFt} + w_{corridor} \cdot v_{CFt} + w_{door} \cdot v_{DCt}, \tag{15}$$

$$v_r = w_{wall} \cdot v_{WFr} + w_{corridor} \cdot v_{CFr} + w_{door} \cdot v_{DCr}, \tag{16}$$

being v_t the linear velocity and v_r the angular velocity, v_{WFt} and v_{WFr} the linear and angular velocities for the Wall Following behavior, v_{CFt} and v_{CFr} the linear and angular velocities for the Corridor Following behavior, and v_{DCt} and v_{DCr} the linear and angular velocities for the Door Crossing behavior.

Weights changed throughout the experiment, depending on the environment configuration, and adapting the robot navigation to its instant situation. All the experiments were carried out for a maximum angular velocity of 15 °/s and a maximum linear velocity of 100 mm/s.

Figure 12(a) presents a simulated environment. It was autonomously explored from the departure point (Start) towards the arrival point (End). The BFR at the end of the exploration process is depicted in Fig. 12b. The three behaviors are merged at each time instant. However, the environment is corridor-like in almost the whole trajectory, as the green color in the BFR proves. Thus, w_{wall} contributes with the highest weighting factor in many areas of the robot navigation and, hence, Corridor Following behavior becomes the main contributor to the motion commands. Nevertheless, such a corridor-like configuration does not match all the robot situations while it navigates. Thereby, close to the departure point, the robot perceives the environment as a wall (red color), since it started to move to the east, then to the north and finally, turned to the west. It happens, as well, when the robot turns from the south corridor to the north. Furthermore, when $w_{corridor}$ is the highest weighting factor, its value is between 0.8 and 0.9 in the most points, even above 0.9 in some points of the trajectory. The BFR

https://doi.org/10.1017/S0263574715000478 Published online by Cambridge University Press

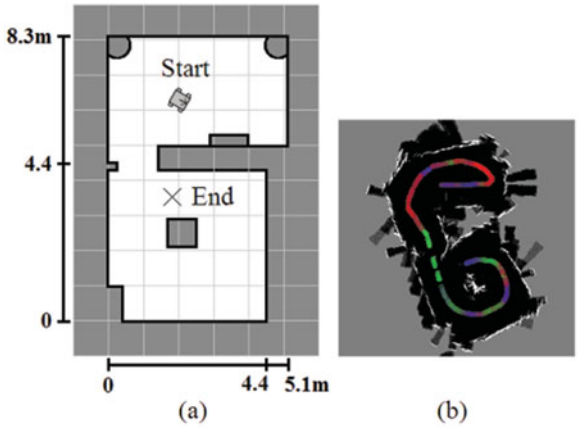


Fig. 13. Real environment: (a) configuration; (b) BFR.

in Fig. 12(b) shows the validity of the method to fuse behaviors, since the weighting factors w_{wall}, $w_{corridor}$ and w_{door}, calculated with the proposed technique, allows to adapt the navigation of the robot to the area being explored. It must also be pointed out that the method yields another advantage, as derived from Fig. 12(b). The weighting factors do not change abruptly, but progressively adapt to changes in the environment configuration. This means that the BFR presents a soft transition between colors, which will lead to softer movements during navigation.

Autonomous navigation tests were also performed in real scenarios. One of such environments is presented in Fig. 13(a). It consists of a 4.4×4.4 m^2 room joined to a 3.1 m wide and 5.1 m long corridor through a 1.1 m door. Its BFR at the end of the exploration process is shown in Fig. 13(b). At the beginning, the knowledge of the environment is very reduced. Hence, although it is expected that w_{wall} to be higher than $w_{corridor}$ and w_{door}, it is not actually true. When the robot is near the right wall of the corridor, the Wall Following behavior becomes the most important (red color). From now on till the door, the robot mainly follows a wall, being w_{wall} the higher weighting factor. Then, the agent perceives the environment as a corridor because it encounters a wall on its left and on its right (green color). When the robot starts to explore the room, the robot turns as if the environment configuration was like a door (blue color). In the room, there is an alternation in the behavior which most contributes to the robot motion commands. Nevertheless, behaviors are merged applying the weighting factors w_{wall}, $w_{corridor}$ and w_{door} to efficiently adapt the robot motion command to the place of the environment it is navigating through.

5. Conclusions

This paper has presented an algorithm to obtain the weighting factors to fuse three reactive behaviors from place characterization for robot navigation. The proposed method reduces a local grid built from sonar sensors to a nine-dimensional short feature vector, which explains the 99.5% of the original information. The technique extracts the contour of the of RFO around the robot. The contour is represented by its depth map, which is processed to obtain its $|FFT|$. Then, the $|FFT|$ if projected onto the base of the system to calculate the nine-dimensional feature vector. The three weighting factors are derived from the comparison of the feature vector with the patterns. One of the advantages of the method is that both the base calculation and the patterns calculation have to be obtained only once and off-line. Both procedures are independent, making the method more flexible and adaptable. Furthermore, the test environments differ from those used to capture the local maps for the base calculation and also differ from the scenarios were the local maps to calculate the nine patterns were acquired.

The time consumption of the method allows its application in an on-line robot navigation task. Tests have proven that the algorithm needs less than 2 ms to determine the weighting factors to fuse the motion commands of the three reactive behaviors. Thus, these results show that the robot could rapidly react to unexpected situations while navigating and, hence, its movement would efficiently adapt to the environment configuration at each time instant.

https://doi.org/10.1017/S0263574715000478 Published online by Cambridge University Press

Results after manually guiding the robot and in an autonomous navigation task, both in simulated and real environments, have proven the validity of the method, since the weighting factors at each point of the robot trajectory are consistent with the environment configuration. Besides, weights do not change abruptly, but progressively adapt to changes in the environment configuration. It must be noted that a graphical representation, the BFR, has been developed to show these results.

References

1. R. C. Arkin, *Behavior-Based Robotics* (MIT Press, Cambridge, USA, 1998).
2. M. J. Mataric, "Behavior-based control: Examples from navigation, learning and group behavior," *J. Exp. Theor. Artif. Intell. Spec. Issue: Softw. Archit. Phys. Agents* **9**(2–3), 323–336 (1997).
3. E. Zalama, J. Gómez, M. Paul and J. R. Perán, "Adaptive behavior navigation of a mobile robot," *IEEE Trans. Syst. Man Cybern.-Part A: Syst. Humans* **32**(1), 160–169 (Jan. 2002).
4. P. Rusu, E. M. Petriu, T. E. Whalen, A. Cornell and H. J. W. Spoelder, "Behavior-based neuro-fuzzy controller for mobile robot navigation," *IEEE Trans. Instrum. Meas.* **52**(4), 1335–1340 (Aug. 2003).
5. D. Gu and H. Hu, "Teaching Robots to Coordinate its Behaviours," *Proceedings of the 2004 IEEE International Conference on Robotics and Automation (ICRA 2004)*, Nueva Orleans, USA (Apr., 2004) pp. 3721–3726.
6. N. Rahman and A. R. Jafri, "Two Layered Behaviour Based Navigation of a Mobile Robot in an Unstructured Environment using Fuzzy Logic," *Proceedings of the IEEE 2005 International Conference on Emerging Technologies*, Islamabad, Pakistan (Sep. 2005) pp. 230–235.
7. E. Freire, T. Bastos, M. Sarcinelli and R. Carelli, "A new mobile robot control approach via fusion of control signals," *IEEE Trans. Syst. Man Cybern.-Parb B: Cybern.* **34**(1), 419–429 (Feb. 2004).
8. S. Parasuraman, V. Ganapathy and B. Shirinzadeh, "Behaviour Based Mobile Robot Navigation Technique for Real World Environments using Fuzzy Logic System," *Proceedings of the 2004 IEEE International Conference on Systems, Man and Cybernetics*, Netherlands (2004) pp. 3359–3364.
9. S. Khatoon and S. Khatoon, "Behavior Coordination of Autonomous Mobile Robot Navigation by Neuro-Fuzzy System," *Proceedings of the IEEE 31st Annual Northeast Bioengineering Conference*, USA (Apr., 2005) pp. 56–60.
10. M. F. Selekwa, D. D. Dunlap and E. G. Collins, "Implementation of Multi-Valued Fuzzy Behavior Control for Robot Navigation in Cluttered Environments," *Proceedings of the 2005 IEEE International Conference on Robotics and Automation (ICRA 2005)*, Barcelona, Spain (Apr., 2005) pp. 3688–3695.
11. A. M. Martínez and J. Vitriá, "Clustering in image space for place recognition and visual annotations for human-robot interaction," *IEEE Trans. Syst. Man Cybern.–Part B: Cybern.* **31**(5), 669–682 (Oct. 2001).
12. Q. V. Do and L. C. Jain, "A Visual Landmark Recognition System for Autonomous Robot Navigation," *Proceedings of the International Conference on Computational Intelligence for Modelling Control and Automation*, Sydney, Australia (2006) pp. 237–242.
13. A. Pronobis, B. Caputo, P. Jensfelt and H. I. Christensen, "A Discriminative Approach to Robust Visual Place Recognition," *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2006)*, Beijing, China (Oct., 2006) pp. 3829–3836.
14. M. Larbi and B. Aek, "Mobile Robot Recognition using Bayesian Penalization with Neural approach," *Proceedings of the 2005 ISSC Congress on Computational Intelligence Methods and Applications*, Istanbul, Turkey (Dec., 2005).
15. O. Martínez, C. Stachniss and W. Burgard, "Supervised Learning of Places from Range Data using Adaboost," *Proceedings of the 2005 IEEE International Conference on Robotics and Automation (ICRA 2005)*, Barcelona, Spain (Apr., 2005) pp. 1730–1735.
16. L. Shi, S. Kodagoda and G. Dissanayake, "Laser Range Data Based Semantic Labeling of Places," *Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2010)*, Taipei, Taiwan (Oct., 2010) pp. 5941–5946.
17. P. Sousa, R. Araújo and U. Nunes, "Real-Time Labeling of Places using Support Vector Machines," *Proceedings of the 2007 International Symposium on Industrial Electronics (ISIE 2007)*, Vigo, Spain (Jun., 2007).
18. O. Martínez-Mozos, H. Mizutani, H. Jung, R. Kurazume and T. Hasegawa, "Categorization of indoor places by combining local binary pattern histograms of range and reflectance data from laser range finders," *Adv. Robot.* **27**(18), 1455–1464 (2013).
19. H. P. Moravec, "Sensor fusion in certainty grids for mobile robots," *AI Mag.* **9**(2), 61–74 (1988).
20. K. Lee, S.-J. Lee, M. Kolsch and W. K. Chung, "Enhanced maximum likelihood grid map with reprocessing incorrect sonar measurements," *Autonomous Robots* **35**(2–3), 123–141 (2013).
21. S.-J. Lee, D.-W. Cho and J.-B. Song, "Novel sonar salient feature structure for extended kalman filter-based simultaneous localization and mapping of mobile robots," *Adv. Robot.* **26**(8–9), 1055–1074 (2012).
22. M. Mucientes, D. L. Moreno, A. Bugarín and S. Barro, "Evolutionary learning of a fuzzy controller for wall-following behavior in mobile robotics," *Soft Comput.* **10**, 881–889 (2006).
23. H. Zhang, S. Liu and S. X. Yang, "A Hybrid Robot Navigation Approach Based on Partial Planning and Emotion-Based Behavior Coordination," *Proceedings of the 2006 IEEE/RSJ Interantional Conference on Intelligent Robots and Systems (IROS 2006)*, Beijing, China (Oct., 2006) pp. 1183–1188.

https://doi.org/10.1017/S0263574715000478 Published online by Cambridge University Press

24. H. Tamini and A. Zell, "Vision Based Localization of Mobile Robots using Kernel Approaches," *Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2004)*, Sendai, Japan (Sep.–Oct., 2004) pp. 1896–1901.
25. C. Cadena, D. Gálvez-López, J. D. Tardós and J. Neira, "Robust place recognition with stereo sequences," *IEEE Trans. Robot.* **28**(4), 871–885 (2012).
26. J. L. Crowley, F. Wallner and B. Schiele, "Position estimation using principal components of range data," *Robot. Auton. Syst.* **23**(4), 267–276 (1998).
27. N. Vlassis, Y. Motomura and B. Krose, "Supervised Linear Feature Extraction for Mobile Robot Localization," *Proceedings of the 2000 IEEE International Conference on Robotics and Automation (ICRA 2000)*, San Francisco, USA (Apr., 2000) pp. 2979–2984.
28. N. I. Giannoccaro and L. Spedicato, "A PCA-Based Interpretation of Ultrasonic Data for Accurately Detecting a Corridor Wall," *Proceedings of the 9th International Multi-Conference on Systems, Signals and Devices*, Chemnitz, Germany (Mar., 2012) pp. 1–6.
29. X. Qu, M. Yao, Q. Gu and J. Zhang, "Adaptive Subspace Base Online PCA Algorithm for Mobile Robot Scene Learning and Recognition," *Proceedings of the 2011 International Conference on Intelligent Human-Machine Systems and Cybernetics*, Hangzhou, China (Aug., 2011) pp. 205–209.
30. A. Brooks, A. Makarenko and B. Upcroft, "Gaussian process models for indoor and outdoor sensor-centric robot localization," *IEEE Trans. Robot.* **24**(6), 1341–1351 (2008).
31. J. D. Holliday, N. Salim, M. Whittle and P. Willett, "Analysis and display of the size dependence of chemical similarity coefficients," *J. Chem. Inform. Comput. Sci.* **43**, 819–828 (2003).
32. A. Poncela, C. Urdiales, E. J. Pérez and F. Sandoval, "A new efficiency-weighted strategy for continuous human/robot cooperation in navigation," *IEEE Trans. Syst. Man Cybern.–Part A: Syst. Humans* **39**(3), 486–500 (2009).

https://doi.org/10.1017/S0263574715000478 Published online by Cambridge University Press