Robotica (2008) volume 26, pp. 99–107. © 2007 Cambridge University Press
doi:10.1017/S0263574707003633 Printed in the United Kingdom

Object learning and detection using evolutionary deformable models for mobile robot navigation

M. Mata[†]*, J. M. Armingol[‡], J. Fernández[†], and A. de la Escalera[‡]

[†]*Computer Architecture and Automation Department, Universidad Europea de Madrid, Villaviciosa de Odon, 28670 Madrid, Spain.*
[‡]*Intelligent Systems Laboratory, Universidad Carlos III de Madrid, Leganés, 28911 Madrid, Spain.*
E-mail: armingol@ing.uc3m.es

(Received in Final Form: June 1, 2007. First published online: October 25, 2007)

SUMMARY
Deformable models have been studied in image analysis over the last decade and used for recognition of flexible or rigid templates under diverse viewing conditions. This article addresses the question of how to define a deformable model for a real-time color vision system for mobile robot navigation. Instead of receiving the detailed model definition from the user, the algorithm extracts and learns the information from each object automatically. How well a model represents the template that exists in the image is measured by an energy function. Its minimum corresponds to the model that best fits with the image and it is found by a genetic algorithm that handles the model deformation. At a later stage, if there is symbolic information inside the object, it is extracted and interpreted using a neural network. The resulting perception module has been integrated successfully in a complex navigation system. Various experimental results in real environments are presented in this article, showing the effectiveness and capacity of the system.

KEYWORDS: Computer vision; Matching learning; Deformable; models; Landmark navigation.

1. Introduction

A mobile robot must have a reliable navigation system for avoiding objects in its path and recognizing important objects of the environment to identify places. A prerequisite for geometric navigation of a mobile robot is a position-finding method. As a mobile robot moves through its environment, its actual position and orientation always differ from the position and orientation that it is commanded to hold, with wheel slippage being a major source of error.[6] The errors accumulate and the localization uncertainty increases over time. Topological navigation allows to overcome some of the classical problems of geometric navigation in mobile robots such as position uncertainty processing and perception of the environment.

On the other hand, topological navigation is heavily dependent on a powerful perception system to identify elements of the environment.[10] Chosen elements should be simple enough in order to permit an easy identification from different view angles and distances. Machine learning can help topological navigation to solve these problems. Its techniques are being applied with remarkable success to several cases of computer vision and perception, but most of these problems are fairly simple in nature and still cannot handle real-time requirements.[2,7,21] The difficulty with scaling up to complex tasks is that inductive learning methods require a very large number of training patterns in order to generalize correctly from high-density sensor information (like video cameras).[16,20,22] Deformable models have been intensively studied in image analysis through the last decade,[9,18] and are used for detection and recognition of flexible or rigid models under various viewing conditions.[4] They have been applied for querying databases given the object shape, color, and texture;[24,25] motion-based segmentation of deformable structures undergoing non-rigid movements through shape and optical flow;[14] for Intelligent Vehicles, they have been used to detect road signs[4,8] and road borders.[15] They are commonly used for human face detection and tracking;[12] recognizing characters and lineal symbols in handwritten line drawings;[23] in medical imagery, they have been used for the segmentation of deep brain nuclei in 3D MRI[17] or human melanoma cancer cells in confocal microscopy imaging.[18]

However, recent results in mobile robot learning have demonstrated that robots can learn simple objects identification from very little initial knowledge information in restricted environments.[5,13,19]

There are two major approaches in the use of landmarks for topological navigation in related literature. One approach uses as landmarks regions of the environment that can be recognized later, although they are not single objects. For example, in Ref. [1], a spatial navigation system based on visual templates is presented. Templates are created by selecting a number of high-contrast features in the image and storing them together with their relative spatial locations in the image. Franz *et al.*[11] have developed a vision-based system for topological navigation in open environments. This system represents selected places by local 360° views of the surrounding scenes. The second approach uses objects of the environment as landmarks, with perception algorithms designed specifically for each object. Beccari *et al.*[3] describe a series of motor and perceptual behaviors used for indoor navigation of mobile robots; walls, doors, and corridors are used as landmarks.

* Corresponding author. E-mail: mmata@uem.es

https://doi.org/10.1017/S0263574707003633 Published online by Cambridge University Press

A real-time color vision-based general object detection and recognition system is presented in this article. The proposed particularized system maintains enough generality to cope with the detection of nearly any planar object in cluttered, uncontrolled real images, and with prerequisite computation times. It uses an effective representation of objects by means of deformable models, and is easily adaptable to detect new objects by training from images, with minimal human intervention (just marking the object to learn in the training images).

This article is organized as follows. Section 2 is dedicated to the details of the practical implementation of the object detection system. The learning process for new objects to detect is covered in Section 3. Some real experimental results showing the performances of the proposed system are included in Section 4, together with a practical analysis of the system's limitations. Finally, Section 5 holds relevant conclusions and the main lines for future works.

2. Landmark Detection System

The proposed two-dimensional (2D) Pattern Search Engine is designed for landmark searching, although it can be used for many other 2D object recognition applications. When an appropriate landmark is found, the symbolic information it can contain (text or icons), is extracted and interpreted using a neural network.

The main problems when trying to detect objects that humans use as landmarks are perspective deformation and illumination. Deformable models are used to handle perspective deformations, and Hue, Saturation, Lightness (HSL) color space and real image-based training cope with illumination. Thus, the proposed method has the advantages over visual templates that it copes easier with the perspective deformations and needs less patterns loaded in memory.

A three-stage algorithm is used (Fig. 1). First, regions of interest (ROIs) are extracted. Then, the extracted ROIs are used to initialize a genetic algorithm (GA) for the landmark search through the image. Each individual of this GA encodes a deformable model. The fitness of an individual is a measure of the matching between the deformed model it encodes and the landmark searched for. Finally, if a landmark is found, symbols are extracted and identified with a classical backpropagation neural network. Keep in mind that the learning step is described in Section 3.

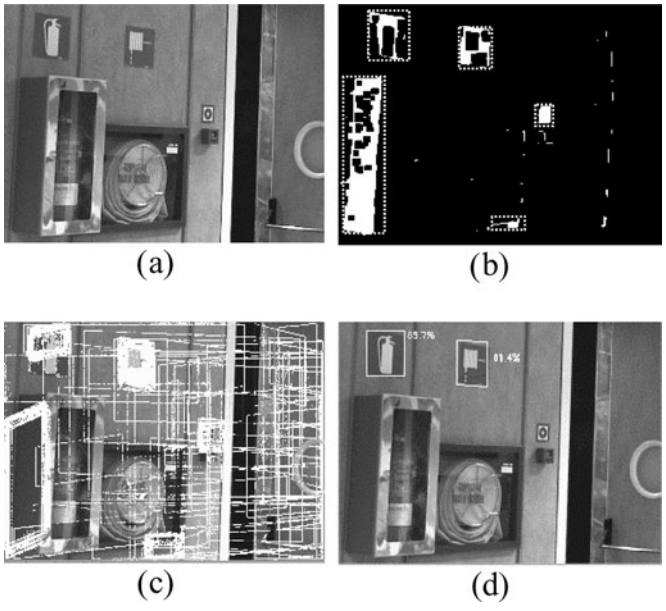


Fig. 1 (a) Original image. (b) ROIs. (c) GA initialization. (d) Detected landmarks.

2.1. Extraction of regions of interest

A quick but powerful color segmentation is done on the target image, in order to establish the ROIs. These are zones where the selected model has a relevant probability of being found. The color interval used for this segmentation is recovered from a database, previously *learned* by the system from training images. In the segmented image, blobs with the appropriate geometry are selected as ROIs. ROIs marked in a real example are shown in Fig. 1(b). The learning strategy has proven very reliable *if* adequate training images are provided (i.e., they must cover *real* situations over the whole application illumination conditions range).

The color segmentation is made in H, S, and L components of the image $\mathbf{I}(x, y)$ separately, and combining them with an AND logical operation, leading to binary image $B(x, y)$:

$$B(x, y) = \begin{cases} 1 & \begin{array}{l}\{H_1 \leq H(x, y) \leq H_2\} \text{ AND} \\ \{L_1 \leq L(x, y) \leq L_2\} \\ \quad \text{AND } \{S_1 \leq S(x, y) \leq S_2\}.\end{array} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

Then, some morphological operations are carried out for connecting disjointed blobs.

On the other hand, Hue presents some drawbacks. First, it is an angular component, so the values 0 and 256 are exactly the same (circular continuity); this must be taken into account when segmenting a color interval. Second, Hue is not defined for low or null values of saturation; in these situations, the pixels are achromatic, and Hue can take erratic values. The first issue is easy to overcome by segmenting in two steps, but the second one requires a more complex treatment. In this work, the 255 value for Hue is reserved and labeled as achromatic. Hue component is rescaled to 0–254, and pixels having low saturation are set to the achromatic value. For the rest of the processes, when a pixel is set as achromatic, only the L component is used for it.

2.2. Landmark search

A GA is used to confirm or reject the ROI hypothesis. Each individual's genome is made up of five variables: the deformed model's cartesian coordinates (x, y) in the image, its horizontal and vertical size in pixels $(\Delta X, \Delta Y)$, and a measure of its vertical perspective distortion (SkewY). Figure 2. (a) shows the geometrical interpretation of an individual, representing a deformed model.

Individual fitness is a measure of the matching between the deformed model and the landmark searched for. For this matching, four small pattern-windows, containing adequate fragments of the landmark searched for, are normalize-correlated with the image, in the position established by the deformed model. A small size of the pattern-windows

https://doi.org/10.1017/S0263574707003633 Published online by Cambridge University Press

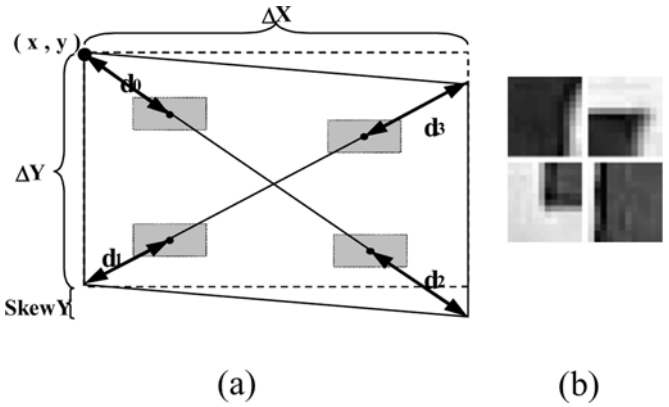


Figure 2. (a) Individual's geometric interpretation. (b) Example pattern-windows.

allows handling a wide interval of perspective deformation with low computational cost, but a big-size window has more information and causes less false positives. Experimentally, a window size of 10×10 pixels presents good relationships between these factors. These pattern-windows are previously learned and stored by the system (Section 3). They are placed over the model diagonals to exploit frequent symmetries in the objects used as landmarks. The relationships between the pattern-windows' diagonals and the model corners' relative distances are kept constant, so the windows follow the model deformations, as shown in Fig. 2 (a). Example pattern-windows are presented in Fig. 2 (b), used with the fire emergency signs in Fig. 1.

The GA is initialized by centering a big part of the total population around the previously established ROI. In order to preserve genetic diversity, the number of individuals for every ROI is a minimum number plus a quantity proportional to each size. A fixed number is randomly distributed around the whole image. This allows working with very few individuals (less than 100) and a quick convergence if the ROI is really a valid hypothesis (one or two generations are usually enough). On the other hand, a false hypothesis makes the population to go away from the real ROI, but as the fitness function would have a low value, it would be rejected in a generation. Finally, the individual's fitness is a good measure of the match certainty. Figure 1 (c) presents GA initialization, and Fig. 1 (d) shows confirmed landmarks after the second GA generation.

Normalized correlation over the L component [Eq. (2)] is used for comparing the pattern-windows $M_k(x, y)$ with the image background $L(x, y)$; these pattern-windows are correlated in the positions fixed by the parameters encoded by the GA individual whose fitness is being evaluated.

$$r_k(x, y) = \frac{\sum_i \sum_j (L(x+i, y+j) - \overline{L}) . (M_k(i, j) - \overline{M_k})}{\sqrt{\sum_i \sum_j (L(x+i, y+j) - \overline{L})^2 . \sum_i \sum_j (M_k(i, j) - \overline{M_k})^2}}. \quad (2)$$

$$\rho_k(x, y) = \max(r_k(x, y), 0)^2.$$

Normalized correlation makes fitness estimation robust to illumination changes, and provides a means to combine local and semiglobal range for the pattern-windows. First, correlation is maximal exactly in the point where a pattern-window is over the corresponding detail of the object in the image, as needed for achieving a precise alignment between the model and the object. Second, the correlation falls down as the pattern-window goes far from the exact position, but it keeps a medium value in a small neighborhood of it; this gives a moderate fitness score to individuals located near an object but not exactly over it, making the GA converge faster.

2.3. Extraction of landmark-associated information

If a new landmark is found in the target image, then the relevant information for the localization process is extracted. If the landmark contains symbolic information, the next step is *reading* its contents. This ability is widely used by humans. By symbols, we mean both alphanumeric characters (text and numbers) and iconic figures (signs and alike).

The inner region is isolated and segmented using an adaptive threshold. Blobs are normalized in size and sent to a classifier (a feed-forward neural network with one hidden layer), which assigns a numeric label to each blob, and provides an estimation of the recognition's confidence. If a symbol is not recognized, it is marked with a question mark "?" and its confidence set to zero. This classical network has proved to have a very good ratio between recognition ability and speed compared to more complex neural networks. This network is trained offline using the Quick Propagation Algorithm. The input layer has 524 neurons, the hidden layer has 150 neurons, and the output layer 39 neurons.

The recognized symbol string (with "?" characters representing the ones with insufficient certainty) is sent to the navigation module. This way the perception system gives the confidence on the extracted information to the control-navigation system of the robot. Figure 3 (a) shows the inner region of an office's nameplate found in a real image. In our environment, the first number represents the building, the second represents the floor, followed by a character depending on the floor's area, and finally a number for the office. This kind of information is needed by the topological localization module of the robot. In Fig. 3 (b), blobs considered as possible characters are shown, and, in Fig. 3 (c), binary size-normalized images, that the neural network has to recognize, are included. In this example, recognition confidence is over 85% for every character.

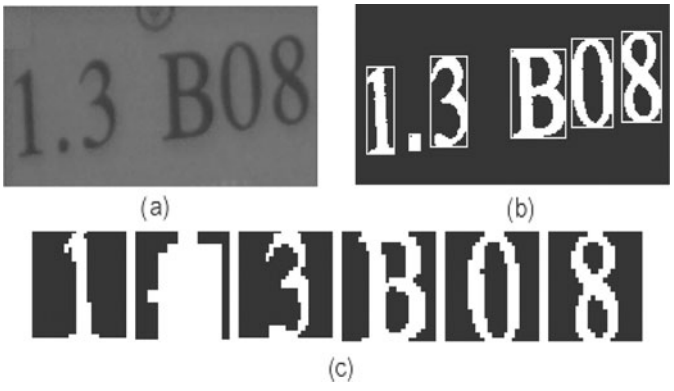


Fig. 3. Symbol recognition stage

https://doi.org/10.1017/S0263574707003633 Published online by Cambridge University Press

3. Learning Recognition's Parameters

For the learning process, the human teacher must provide several training images, where the landmark(s) to learn are bounded by rectangular boxes (these boxes will be referred to as *target boxes* from now on). There are no *a priori* restrictions for the set of training images provided. Of course, the wider the conditions this set of images covers (illumination, background, perspective distortions, etc.), the best results will obtain the learned parameters on real situations. In this case, an average number of 30 images per landmark have been used.

As previously established, the recognition process can be sequentially divided into two steps: candidate hypotheses generation (through segmentation) and hypotheses verification (using correlation). Consequently, the learning process for a new landmark is divided into two stages. In the first step, thresholding levels for segmentation are found. The second step is dedicated to determine the location of the correlation-windows inside one individual.

3.1. Learning segmentation parameters

In the first training step, the system has to learn the best threshold values for the color segmentation of the landmark. Upper and lower thresholds for HSL components must be estimated. These six values will confirm the genome of the individuals of a GA that will search through the training image space.

The fitness function for the GA must encourage that the segmented regions each individual generates match the target boxes, defined in the training images. On the other hand, there should not appear segmented regions outside the target boxes, if possible. The ideal segmentation result should be a binary black image with the target boxes' corresponding zones in white. This "ideal image" can be matched with the binary image resulting from the individual's genome, using an XOR logical function. Pixels that survive this operation are missclassified, since they have been included in the segmented regions while they should not have been (or its inverse). The count of white pixels after the XOR pass is then a good measure of the segmentation error for the considered training image. The total segmentation error for one individual is obtained by repeating this operation for all the training image sets and accumulating the missclassified pixels in each image:

$$E = \frac{1}{N}\sum_{n=0}^{N}\left[\sum_{x=0}^{X}\sum_{y=0}^{Y}\left(B_T^n(x,\ y)\,\mathrm{XOR}\,B^n(x,\ y)\right)\right] \qquad (3)$$

where B_T^n is the "ideal image" and B^n is the binary image resulting from an individual's genome.

The fitness function is then chosen as an inverse function of this total error.

Before the learning run, a coarse initialization of the GA is done, in order to decrease search time. A set of initial H, L, and S threshold values is obtained from any of the training images, using local histograms. Two histograms are computed for each component from the inner and outer regions adjacent to the target box's boundaries (Fig. 4). The inner histograms contain information from the landmark, the background, and noise. The outer histograms contain information from the background, other objects, and noise. For each component, its outer histogram is subtracted from the corresponding inner histogram, with negative values forced to zero. The resulting difference histogram (denoted LDH, from Local Difference Histogram) will contain only information belonging to the desired landmark and not present in the outer region. Finally, initialization values for thresholding are taken from a peak search over the LDH (various peaks are selected for each component).

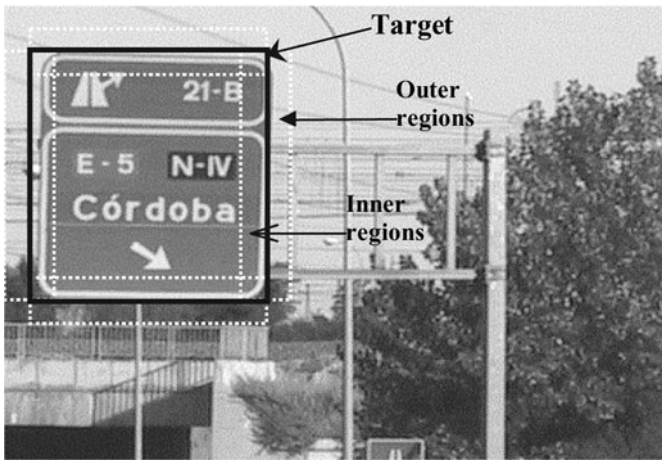


Fig. 4. LDH regions.

This way several values for H, L, and S thresholds are estimated, and their possible combinations generate a good part of the GA's initial population. The rest of the population is randomly generated. This initialization speeds up the training process considerably (Fig. 5).

3.2. Learning correlation windows

The second training step deals with the location of the four correlation-windows inside the individual. Once an individual is situated over a ROI (or a target box in a training image), the problem translates to find the adimensional values $d_0, \ldots, d_3$ that determine the best position for the correlation-windows (see Fig. 2). Again, a GA is used to find these four values (that will comprise each individual's genome).

The correlation-windows should be chosen so that each of them has a high correlation value one and only one location inside the target box, and low correlation values outside it. The selected fitness function for an individual is a positive function of each window's correlation in a small neighborhood (3–5 pixels) of the theoretical position of the window's center (given by the corresponding d_i value from

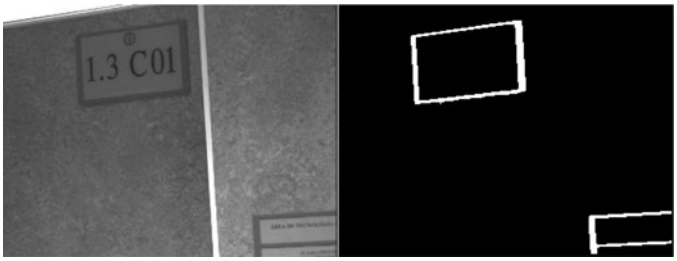


Fig. 5. Learned segmentation example (room information panel).

https://doi.org/10.1017/S0263574707003633 Published online by Cambridge University Press

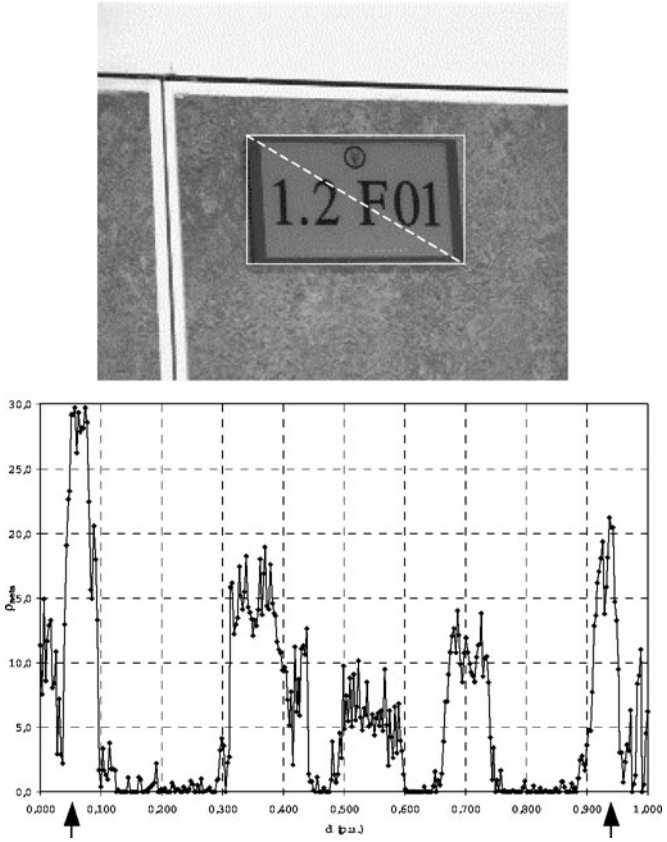


Fig. 6. Estimation of values for d_i.

the individual's genome). This small zone will be referred to as the "theoretical zone." On the other hand, the fitness function includes negative terms counting the maximum correlation of each window inside the target box (but outside the theoretical zone) and the maximum correlation in random zones outside the target box.

Once again, a coarse GA initialization can be easily done in order to decrease training time. Intuitively, the relevant positions where the correlation-windows should be placed are those having strong local variations in the image components (H, L, and/or S). A simple method is used to find locations like that. The diagonal lines of the diagonal box of a training image (which will match those of a theoretical individual) are scanned to H, L, and S vectors. Inside these vectors, a local estimate of the derivative is calculated. Then, positions having a high local derivative value are chosen to compute possible initial values for the d_i parameters. Figure 6 shows this process, where the plot represents the derivative estimation for the marked diagonal (starting from the top left corner), while the vertical bars over the graph indicate the selected initial d_i values.

This function provides a measure for each d_i value; it is evaluated along the diagonals for each target box, and averaged through all target boxes and training images provided, leading to a goodness array for each d_i value. Example pattern-windows selected for some objects are shown (zoomed) in Fig. 7; its real size in pixels can be easily appreciated.

The fitness function F used here is then a function of the normalized correlation of each pattern-window ρ_k,

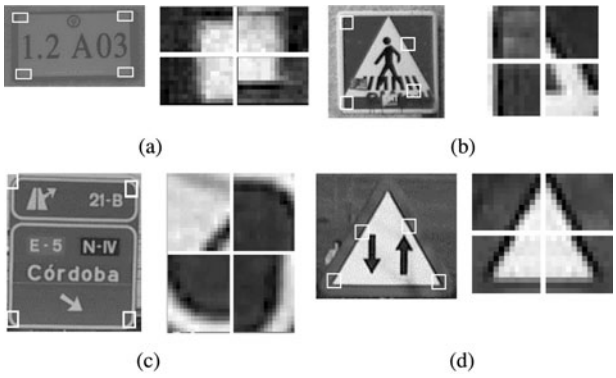


Fig. 7. Learned pattern-windows for some objects. (a) Room informative panel. (b) Pedestrian crossing traffic sign. (c) Road information panels. (d) Triangular traffic signs.

$(0 < \rho_v < 1)$, placed over the image points. It has been empirically tested, leading to the function in Eq. (4)

$$E = \frac{3 - \rho_0.\rho_2 - \rho_1.\rho_3 - \rho_0.\rho_1.\rho_2.\rho_3}{3} \tag{4a}$$

$$F = \frac{1}{0.1 + E} \tag{4b}$$

The error term E in Eq. (4a) is a measure of the difference between the object and the deformed model. It includes a global term with the product of the correlation of the four pattern-windows, and two terms with the product of correlations of pattern-windows in the same diagonal. These last terms force the deformed models to match the full extent of the object, and avoid matching only a part of it. Note that these terms can have low values, but will never be zero in practice, because correlation never reaches this value. Finally, the fitness score in Eq. (4b) is a bounded inverse function of the error. This fitness function allows the system to recognize the landmark although part of its corner is occluded by another object.

The receiver operating characteristic (ROC) curve is frequently used to evaluate performance by plotting the percentage of true positives detected as a function of the false alarms. A higher curve indicates better performance.

The ROC curves shown in (Fig. 8) endorse the four pattern-windows used in the learning correlation process in order to develop a running real-time algorithm.

4. Experimental Results

Within the Systems Engineering and Automation Department in Carlos III University, an advanced topological navigation system has been developed for indoor mobile robots. It uses a laser telemeter for collision avoidance and door-crossing tasks, and a color vision system for high-level localization tasks (Fig. 9). It uses deformable models for locating landmarks of the environment, and extracts information (including text and icons) that is used for navigation control.

Experiments have been carried out in the Department premises. It is a typical office environment, with corridors, halls, offices, and some large rooms (Fig. 10).

https://doi.org/10.1017/S0263574707003633 Published online by Cambridge University Press

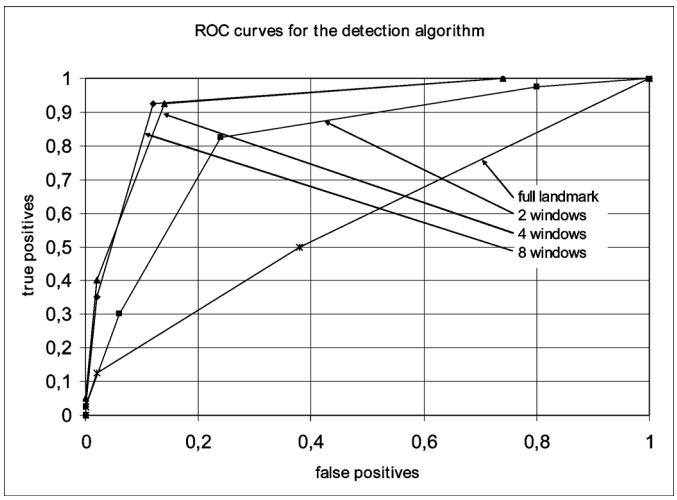


Fig. 8. ROC curves for the detection algorithm as a function of the pattern-windows used.

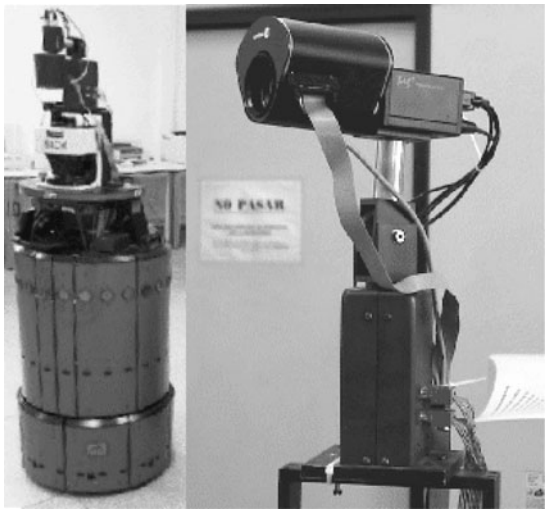


Fig. 9. RWI B-21 test robot and artificial vision system.

Each of the floors of the buildings in the campus is organized in zones named with letters (Fig. 3 and 11). Within each zone, rooms and offices are designated with a number. There are office nameplates located at the entrance of room's doors. These landmarks are specially useful for topological navigation for two reasons:

- they indicate the presence of a door. If the door is opened, it is easily detected with the laser telemeter, but it cannot be detected with this sensor when it is closed. So, the detection of the nameplate handles this limitation.
- the system is able to *read* and understand the symbolic content of the landmarks, shown in Fig. 11. This allows an exact "topological localization," and also confirms the detection of the right landmark.

When the *office nameplates* are available, they offer all the information needed for topological navigation. When not, the rest of the landmarks are used. Also, there are other "specially relevant" landmarks: those alerting to the presence of *stairs* or *lifts*, since they indicate the ways of moving to another floor of the building. Finally, emergency exit signs indicate ways of exiting the building.

Thinking of these examples, it should be noted that some landmarks can be used in two ways. First, their presence or absence is used for robot localization in a classic manner. Second, the *contents* of the landmark give high-level information, which is naturally useful for topological navigation, as mentioned earlier. This is allowed by the symbol reading ability included in our landmark detection system. The experimental results will show its usefulness.

In order to show the potential of the proposed system, several navigation experiments are being carried out.

The robot starts in an unknown point of the building, and it must reach a specific location. In this example, the robot starts in the hall between zones B and C, on the third floor of building 1 (see Fig. 12). The robot does not know any of this, and is told to reach room 1.2D01. Figure 12 presents the landmark distribution and the *approximate* trajectory (there is no need for odometric measures) described by the robot.

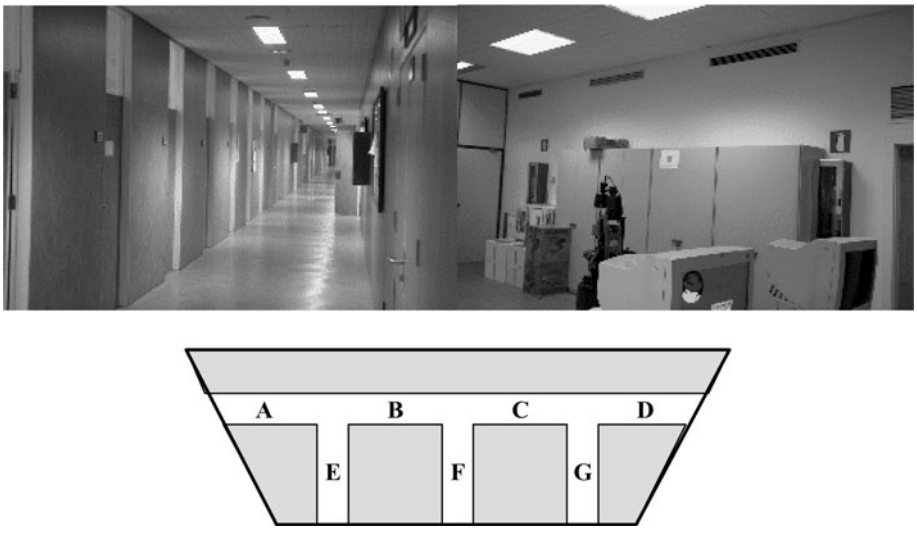


Fig. 10. Environment used in the experiments.

https://doi.org/10.1017/S0263574707003633 Published online by Cambridge University Press

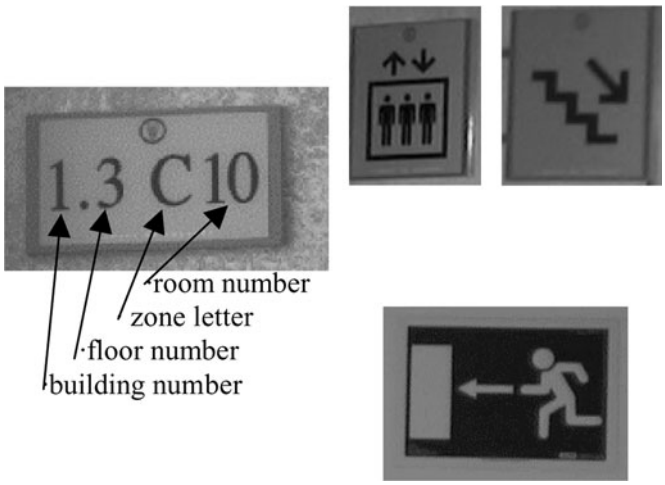


Fig. 11. Some relevant landmarks.

The robot does not know its initial position, so it tries to "find and read a room nameplate landmark." If it can achieve this, then it immediately knows its position (building, zone, and office it stands at). In this case, if it can not find any one, then the "room identification from landmark signature" ability is used. The robot turns around and tries to find all the landmarks around it, and compares the obtained landmark sequence with stored ones. Figure 13. (a) shows an image of this location, taken with the robot's camera. In this example, again this is not enough, because there are several halls with a very similar landmark signature. The last strategy considered by the robot is entering a corridor (using the laser telemeter) and trying again to read a nameplate. Now this is successful, and the robot reads "1.3C01" in the image shown in Fig. 13. (b) Once located, the desired action sequence is generated until the objective room is reached. The robot is in the right building, but in the third floor, so it must search for a lift to go down one floor. The topological map indicates that it has to follow the C-zone corridor, then enter a hall, and search here for a "lift" sign. It follows the corridor, and tries to read the nameplates for avoiding getting lost. If some are missed, it is not a problem, since reading any of the following ones relocates the robot. If desired, other landmarks present in the corridors (like fire extinguisher ones) can be used as an additional navigation aid.

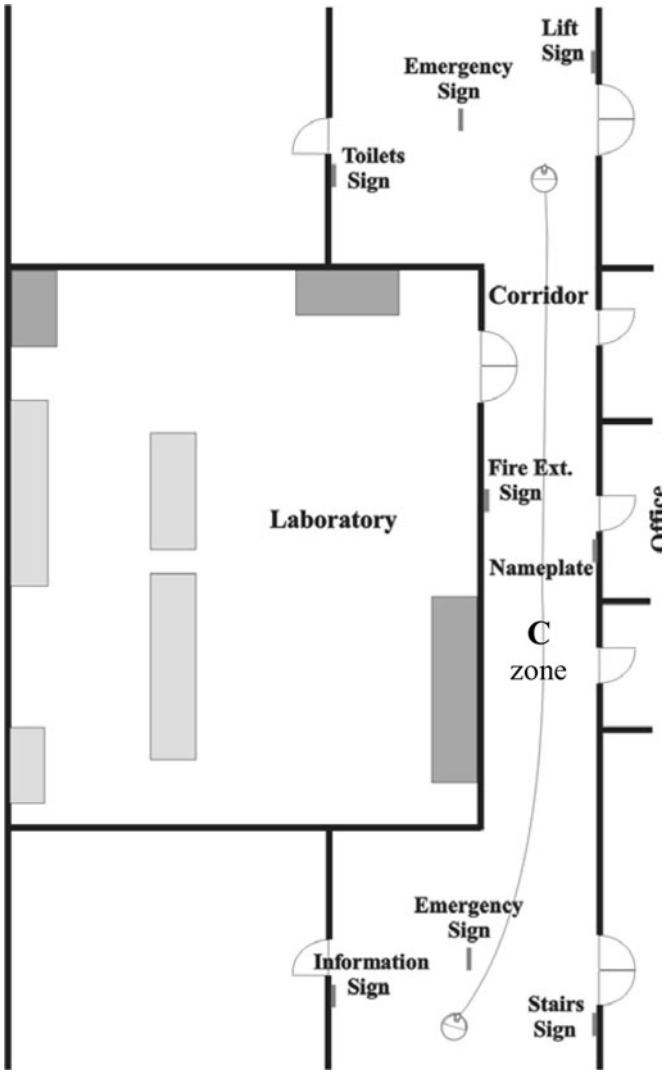


Fig. 12. Navigation test 1.

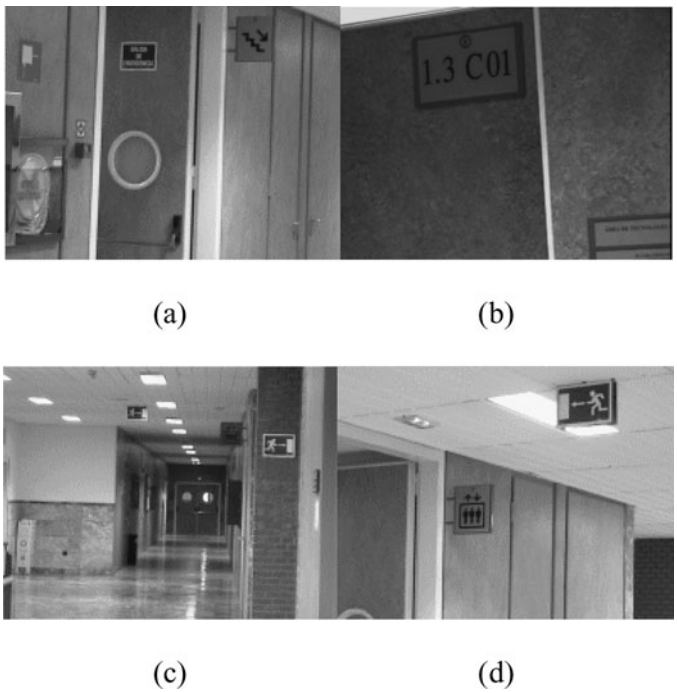


Fig. 13. Some frames in the robot's path.

When the corridor ends in a new hall [Fig. 13 (c)], the robot launches the room identification ability to confirm that. The hall's landmark signature includes the lift sign. When this landmark is found and read [Fig. 13 (d)], the robot finishes its path in this floor, and knows that entering the lift lobby is the way to the second floor. The robot is not able to use the lifts, so the experiment ends here.

A more complex situation is tested in a second part of the experiment. The robot is initially headed, so it will start moving in the wrong direction (entering zone B instead of C, see Fig. 14). When the robot reads the first nameplate in B zone ("1.3B12"), it realizes the wrong direction and heads back to the C-zone corridor, and then follows it like before. Furthermore, this time several landmarks (including the lift one) have been occluded for test purposes. The robot

https://doi.org/10.1017/S0263574707003633 Published online by Cambridge University Press

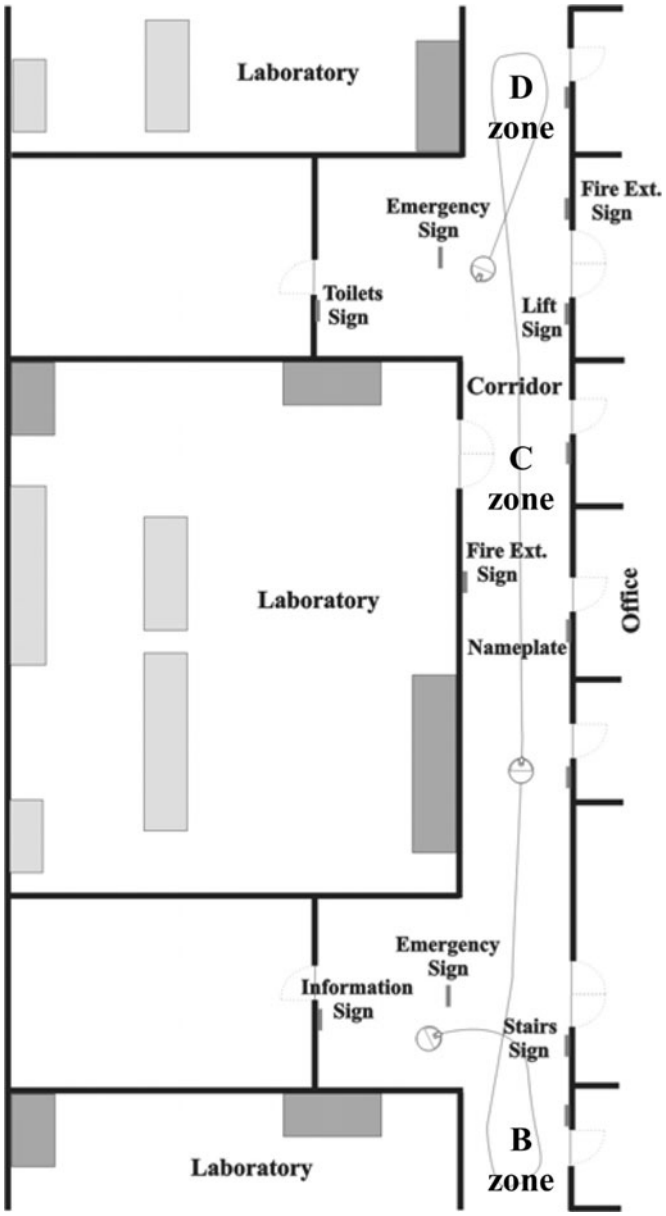


Fig. 14. Navigation test 2.

cannot recognize the hall, so it heads for the new corridor, corresponding to the D zone. When a nameplate is read, the robot knows it has just passed the desired hall and heads back for it. The experiment ends when the robot assures it is in the right hall, but unable to find the occluded lift sign.

5. Conclusions

This article presents a high-level topological navigation application based on a visual landmark recognition system. Its relevant characteristics (learning capacity, generality, and text/symbols reading ability) are exploited for two different tasks. First, "room identification from inside" is achieved through the landmark signature of the room. This can be used for locating the robot without any initialization, and for distinguishing known or new places during map generation tasks. The second example task is searching for a specific place when following a corridor, using the room nameplates placed there for human use, without any information about distance or location of the room. The textual content of the nameplates is read and used to take high-level control decisions. The use of preexistent, human-use designed landmarks, results in a higher degree of integration of mobile robotics in everyday life.

The system has been developed and tested using real indoor and outdoor images, and several example objects have been learned and detected. Field experiments have proven the robustness of the system for illumination conditions and perspective deformation of objects, and applicability limits have been explored. Potential application fields are industrial and mobile robotics, driving aids, and industrial tasks. Its learning ability allows easy application to new and different environments. Actually, it is being used for topological navigation of an indoor mobile robot and for a driver assistance system.

References

1. C. Balkenius, "Spatial learning with perceptually grounded representations," *Robot. Auton. Syst.* **25**, 165–175 (1998).
2. S. M. Bhandarkar, J. Koh and M. Suk, "Multiscale image segmentation using a hierarchical self-organizing map," *Neurocomputing* **14**, 241–272 (1997).
3. G. Beccari, S. Caselli and F. Zanichelli, "QualiSpatial learning with perceptually grounded representations," *Robot. Auton. Syst.* **25**, 165–175 (1998).
4. M. Betke and N. Makris, "Recognition, resolution, and complexity of objects subject to affine transformations," *Int. J. Comput. Vis.* **44** (1), 5–40 (2001).
5. B. Ran, H. X. Liu and W. Martonov, "A Vision-Based Object Detection System for Intelligent Vehicles," *Proceedings of the SPIE—the International Society for Optical Engineering* (1998) vol. 3525, pp. 326–337.
6. J. Borenstein and L. Feng, "Measurement and correction of systematic odometry errors in mobile robots," *IEEE Trans. Robot. Autom.* **12** (5), 869–880 (1996).
7. D. Burschka, J. Geiman and G. Hager, "Optimal Landmark Configuration for Vision-Based Control of Mobile Robots," *Proceedings of the International Conference on Robotics and Automation* (2003) pp. 3917–3922.
8. A. de la Escalera, J. M. Armingol and M. Mata, "Traffic sign recognition and analysis for intelligent vehicles," *Image Vis. Comput.* **11** (3), 247–258 (2003).
9. A. de la Escalera, J. M. Armingol, J. M. Pastor and F. J Rodríguez, "Visual sign information extraction and identification by deformable models for intelligent vehicles," *IEEE Trans. Intell. Transp. Syst.* **5** (2), 57–68 (2004).
10. G. N. DeSouza and A. C. Kak, "Vision for mobile robot navigation: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.* **24** (2), 237–267 (2002).
11. M. O. Franz, B. Schölkopf, H. A. Mallot and H. Bülthoff, "Learning view graphs for robot navigation," *Auton. Robots* **5**, 111–125 (1998).
12. Y. Lijun and A. Basu, "Integrating active face tracking with model based coding," *Pattern Recognit. Lett.* **20** (6), 651–657 (1999).
13. S. Mahadevan and G. Theocharous, "Rapid concept learning for mobile robots,"*Mach. learn.* **31**, 7–27 (1998).
14. C. Kervrann and F. Heitz, "Statistical deformable model-based segmentation of image motion," *IEEE Trans. Image Process.* **8** (4), 583–588 (1999).
15. C. Kreucher and S. Lakshmanan, "LANA: A lane extraction algorithm that uses frequency domain features," *IEEE Trans. Robot. Autom.* **15** (2), 343–350 (1999).

https://doi.org/10.1017/S0263574707003633 Published online by Cambridge University Press

16. S. Marsland, U. Nehmzow and T. Duckett, "Learning to select distinctive landmarks for mobile robot navigation," *Robot. Auton. Syst.* **37**, 241–260 (2001).
17. F. Poupon, J. F. Mangin, D. Hasboun, C. Poupon, I. Magnin and V. Frouin, "Multi-object Deformable Templates Dedicated to the Segmentation of Brain Deep Structures," *Proceedings of the Medical Image Computing and Computer Assisted Intervention, First International Conference* (1998) pp. 1134–1143.
18. H. Rue and O. K. Husby, "Identification of partly destroyed objects using deformable templates," *Stat. Comput.* **8** (3), 221–228 (1998).
19. P. Sala, R. Sim, A. Shokoufandeh and S. Dickinson, "Landmark selection for vision-based navigation," *IEEE Trans. Robot.* **22** (2), 334–349 (2006).
20. S. Se and D. G. Lowe and "Mobile Robot Localization and Mapping with Uncertainty using Scale-Invariant Visual Landmarks," *Int. J. Robot. Res.* **21** (8), 735–758 (2002).
21. S. Se, D. G. Lowe and J. J. Little, "Vision-based global localization and mapping for mobile robots" *IEEE Trans. Robot.* **21** (3), 364–375 (2005).
22. H. Tamimi and A. Zell, "Vision Based Localization of Mobile Robots Using Kernel Approaches," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* (2004) pp. 1896–1901.
23. E. Valveny and E. Marti, "Application of Deformable Template Matching to Symbol Recognition in Handwritten Architectural Drawings," *Proceedings of the Fifth International Conference on Document Analysis and Recognition* (1999) pp. 483–486.
24. K. Yoon, G. Jang, S. Kim and I. Kweon, "Color landmark based self-localization for indoor mobile robots," *J. Control Autom. Syst. Eng.* **7** (9), 749–757 (2001).
25. Z. Yu and A. K. Jain, "Object localization using color, texture and shape," *Pattern Recognith.* **33** (4), 671–684 (2000).

https://doi.org/10.1017/S0263574707003633 Published online by Cambridge University Press