Natural Landmark Detection for Visually-Guided Robot Navigation

Enric Celaya, Jose-Luis Albarral, Pablo Jiménez, and Carme Torras

Institut de Robòtica i Informàtica Industrial (CSIC-UPC). Barcelona, Spain
{celaya, albarral, jimenez, torras}@iri.upc.edu

Abstract. The main difficulty to attain fully autonomous robot navigation outdoors is the fast detection of reliable visual references, and their subsequent characterization as landmarks for immediate and unambiguous recognition. Aimed at speed, our strategy has been to track salient regions along image streams by just performing on-line pixel sampling. Persistent regions are considered good candidates for landmarks, which are then characterized by a set of subregions with given color and normalized shape. They are stored in a database for posterior recognition during the navigation process. Some experimental results showing landmark-based navigation of the legged robot Lauron III in an outdoor setting are provided.

1 Introduction

Indoor robot navigation has received a great deal of attention, and many of the proposed approaches are now successfully used in industrial settings and other specific applications like hospital couriers or museum guides. Such applications are often strongly dependent on known structured features present in each specific environment [1]. Currently the research interest is rapidly shifting towards service robots able to work in cooperation with humans in more general situations not especially well suited for robot operation. Often, robots are required to navigate in unknown and unstructured outdoor environments, where little assumptions can be made about the kind of objects or structures that can be used for robot guidance.

In such outdoor applications, partly teleoperated robots able to reach autonomously a destination marked by a human operator on the image as seen by the robot are agreed to be very handy. Then, updating the target as the robot advances, permits long-range journeys. Our visually-guided navigation approach provides this performance by relying on natural landmarks.

Most related works use point-based visual features, leading to large landmark databases and high numbers of lookups to attain localization. These drawbacks are palliated by recurring to more involved feature detectors, such as the scale-invariant feature transform (SIFT) [2], and by selecting a maximally-informative subset of landmarks [3]. We explore the alternative approach of relying on only a few region-based features, similarly to [4, 5]. The former of these previous works

builds an environmental model incrementally by using color and stereo range information, while the latter uses a multi-resolution visual attention mechanism to extract image regions most salient in terms of color contrast.

Saliency detection in our system is also based on color, but instead of processing entire static frames, our algorithm can be applied directly to the dynamic image stream acquired while the robot moves. Thus, it can be thought as implementing a form of visual memory that replicates the phenomenon of the persistence of images in the retina of the animals' eyes. Salient regions are used as a filter for the search of visual landmarks in the image.

The paper is structured as follows: Section 2 describes our visually-guided navigation context in which the landmark detection system is to be used. Section 3 introduces our approach to natural landmark detection and identification, showing some results for real images taken by a mobile robot in an outdoor setting. Finally, some conclusions and future work are pointed out in Section 4.

2 An Approach to Visually-Guided Navigation

We developed a visually-guided navigation system [6] in which a user controls the robot by signaling the navigation target in the images received from a camera transported by the robot. This form of navigation control is convenient for exploration purposes or when there is no previous map of the environment, situations in which systems like GPS, even if available, become useless and can be discarded. The user can decide the next target for the robot and change it as new views of the environment become available.

Fig. 1 shows the two main windows of the navigation interface: The *Camera Window* and the *Robot Control Interface*. In the Camera Window, the user can see the images taken from the camera of the robot and control its gaze direction to observe the environment. The user may select the current navigation target by clicking on the image with the mouse.

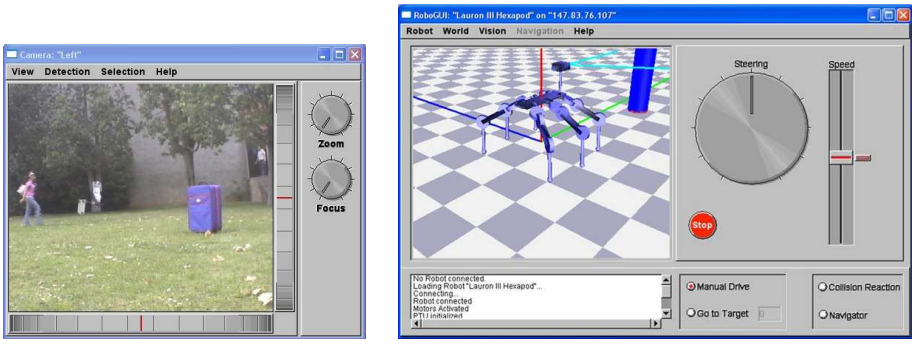

Fig. 1. Camera window and Robot Control Interface.

One of the problems we had to face while designing our visually-guided navigation system was how to specify the navigation target. Ideally, what we wanted is to allow the user to select any object in sight and use it as the current target. The problem with this approach is that, what can be a clearly identifiable object for the user, may not be easily recognized by an artificial vision system, so that it can be lost very soon. Another difficulty that appears with the definition of the target by pointing at it consists in determining what exact area of the image has to be considered as the target object. Our strategy to solve these problems consists in limiting the selection of possible targets to those that the robot is able to identify with its visual recognition system. Thus, it is the system that first shows to the user the set of available landmarks, and the user may select one of them as the target.

Additionally, landmarks detected by the visual system can be used by a landmark-based navigation system (like e.g., that of [7]) to plan a feasible path to the goal in those cases in which obstacles in the way to the goal do not allow a direct approach to it.

3 Natural Landmarks

We relied on the assumption that useful landmarks must be salient, i.e., they must constitute distinctive regions in the image, so that its repeated detection and identification is facilitated. The saliency of a region is not determined by the absolute value of any intrinsic magnitude, but rather by the contrast or difference of this value with respect to the value of the same magnitude in the surroundings [8]. This is also known as *opponency* [9].

Many variables can be used to define the saliency of a region, like color components, intensity, or feature orientation. Works like [10] compute saliency as a combination of the opponency values of these three variables. In what follows, we present a very simple approach that just considers RGB color values, but the same idea could be used with more informative features.

3.1 Detecting Salient Regions

We take an approach to image processing that is inspired in the visual system of living organisms. A key difference between natural eyes and artificial vision systems is that in the eye, individual light receptors fire asynchronously, giving rise to a continuous image flow that can not be naturally decomposed into separate individual image shots. In contrast, artificial systems work in a strictly sequential way taking one frame after another, each involving all individual receptors updated at fixed frequency. Our approach emulates the asynchronous process of the eye by taking pixels at random from the image with a frequency which is not related with that of frame acquisition. Thus, the input image is treated as a continuously varying source of information with no distinction between successive frames.

One advantage of this approach is that the process is independent of the frame rate of the acquisition system, which can be safely ignored in the image processing task. This avoids any need of synchronization between image acquisition and image processing. Another advantage is that the processing cost is independent of image resolution: the same number of pixels per second will be processed whatever the number of pixels of the image is. Image size or resolution affects only the probability that a given pixel is examined in a given period of time.

To detect saliency, the information of individual pixels is used only statistically and stored in local units that cover different regions of the image. This approach allows us to implement a form of visual memory that replicates the phenomenon of the persistence of images in the retina: Units can persist as long as new pixels keep them active, decreasing in strength until disappearing if they are not fed enough by appropriate inputs. The system is robust to sporadic noisy frames, since most units will not be substantially perturbed by noisy pixel values and will keep the essential information through the next uncorrupted frame.

Units have elliptic receptive fields in the image plane, which adapt through successive updates to cover regions whose pixels have similar colors. These ellipses arise naturally from considering normal distributions for pixels taken at random from each region: the center $\mathbf{U}_{xy} = (U_x, U_y)$ of the ellipse corresponds to the mean value of the pixels' positions, whereas its dimensions (the major and the minor axis) and orientation are determined by the covariances. Each unit also has a spherical receptive field in the RGB color space with a fixed radius, which is a parameter of the system, and whose center adapts to approach the average color value of the input pixels to which the unit responded.

Each unit U is defined with the following attributes:

- Center vector $\mathbf{U}_{xy}$ and covariance matrix Σ_{XY} in the image plane,
- Center vector $\mathbf{U}_{rgb}$ in the color space,
- Contrast C_U, a scalar that measures saliency, as explained in point 2) below,
- Creation date $Creac_U$, to record the time from which the unit exists,
- Counter $Updates_U$ for the number of times a unit has been updated,
- Counter $Inside_U$ for the number of times an input pixel has fallen inside the receptive field of the unit in the image plane, and
- Strength S_U, a scalar that estimates the current proportion between the number of pixels to which the unit responds and those lying into the unit's receptive field in the image plane.

Description of the Saliency Detection Algorithm. In the main loop of the algorithm, a random input pixel I is selected and, for each unit U, its Mahalanobis distance in the image coordinates and Euclidean distance in color space are computed as:

$$Mdist_{xy}(U, I) = \sqrt{(\mathbf{I}_{xy} - \mathbf{U}_{xy})^\top \Sigma_{XY}^{-1} (\mathbf{I}_{xy} - \mathbf{U}_{xy})}, \tag{1}$$

where Σ_{XY} is the covariance matrix of unit U, and

$$Edist^2_{RGB}(U, I) = \sum_i (\mathbf{I}_i - \mathbf{U}_i)^2, \qquad \text{with } i \in \{r, g, b\}. \tag{2}$$

The Mahalanobis distance is used in image space instead of the Euclidean one in order have a geometric proximity measure taking into account the shape of the spatial distribution.

If these distances are below given thresholds (MAXDISTXY and MAXDISTRGB, respectively), then the unit is said to respond to the input pixel. Among the units that respond, the one which is closest in the image space, is considered as the *winner*.

1) Updating the winner unit:

The winner is updated according to the following rules:

– **Center position** The unit is approached to the pixel, in the image plane as well as in color space:

$$\mathbf{d}_i = \mathbf{I}_i - \mathbf{U}_i \tag{3}$$

$$\mathbf{U}_i \leftarrow \mathbf{U}_i + \gamma \mathbf{d}_i \tag{4}$$

$$\text{where } i \in \{x, y, r, g, b\} \text{ and } 0 < \gamma < 1$$

– **Covariances in image space** The update of the covariances can be viewed as a simultaneous update of the dimensions and the orientation of the ellipse that represents the unit. The updating of covariances is straightforward:

$$\sigma_{ij} \leftarrow \sigma_{ij} + \gamma(\mathbf{d}_i \mathbf{d}_j - \sigma_{ij}), \qquad \text{where } i, j \in \{x, y\}. \tag{5}$$

– **Counter of updates**

$$Updates_U \leftarrow Updates_U + 1 \tag{6}$$

2) Updating other units:

If the Mahalanobis distance from the input pixel to a non-winner unit is below three times the MAXDISTXY value, the pixel is considered to lay in the unit's neighborhood, and the pixel color is used to update the unit contrast. The update rule is:

$$C_U \leftarrow \alpha C_U + (1 - \alpha) \sqrt{\sum_{i \in \{r,g,b\}} (\mathbf{I}_i - \mathbf{U}_i)^2}, \qquad 0 < \alpha < 1 \tag{7}$$

i.e., increasing or decreasing according to the Euclidean distance in the color space between the input pixel and the unit.

The updating of the strength S_U and $Inside_U$ is done for all units for which the Mahalanobis distance of the input pixel is below the MAXDISTXY value,

i.e., the pixel is in the receptive field of the unit. While $Inside_U$ is simply incremented by one, the strength is increased when the unit responds to the input (i.e., also chromatically) according to:

$$S_U \leftarrow \beta S_U + (1 - \beta), \qquad \text{with } 0 < \beta < 1. \tag{8}$$

If the unit does not respond to the input color, S_U is decreased according to:

$$S_U \leftarrow \beta S_U \tag{9}$$

3) Reallocation of units:

To avoid a proliferation of useless units, the maximum number of them is limited by a parameter of the system, which may be adjusted depending on image complexity and the intended level of detail of the result. When the maximum number of units is reached, in order to allow the creation of new ones corresponding to interesting regions not yet captured by any unit, older ones must be removed. When this is the case, the less useful unit is selected according to the following criteria: First, the unit with the lowest strength value is sought. If its strength is below a certain value, it is assumed that the region it is representing is no longer there, and the unit can be reallocated for the new input. If no low-strength unit is found, the reallocation will only take place provided a contrast estimation of the unit to be created is above that of the lowest-contrast unit. Such contrast estimation is given by the distance in color space of the input pixel to its spatially closest unit.

New units are initialized with center vectors given by the values of the input pixel, with a circular shape in the image plane, and a radius equal to its distance to the closest unit.

4) Merging of units:

When two units respond to a given pixel, they are probably representing different parts of the same region and have to be merged together. For the merged unit, the center values and covariances are computed as a weighted sum of those of the original units. The weights are proportional to the respective area and strength, thus roughly corresponding to the "mass", or number of pixels each unit responds to:

$$Weight(U) = Area(U) S_U \tag{10}$$

The area is computed from the covariances as

$$Area(U) = \sqrt{((\sigma_{xx} + \sigma_{yy} + \Delta) \cdot (\sigma_{xx} + \sigma_{yy} - \Delta))} \tag{11}$$
$$\text{where } \Delta = \sqrt{(\sigma_{xx} - \sigma_{yy})^2 + 4\sigma_{xy}^2}$$

The contrast of the merged unit is set to the highest contrast value of the two original ones, and for the strength, the weighted sum of the strengths is made, this time using the respective values of $Updates_U$ as weights.

5) Output of the system:

As for the output of the system, less relevant units are filtered out before being output as salient regions. To this end, units covering too large regions of the image are discarded, as they usually capture the background and are not useful for navigation. Regions that are too small are also discarded to remove isolated pixels or noise. Finally, units that have not been updated a minimum number of times are not considered, since they are still not reliable enough. From the remaining units, those with contrast values above a given threshold are selected for output as corresponding to salient regions.

3.2 Landmark Characterization

Salient regions are not considered as landmarks by themselves, but as easily recognizable pointers potentially denoting the presence of a landmark, which usually will present a richer structure than a single uniform region. We define a landmark as a set of uniform color regions, each of them with a characteristic color and shape.

Each region R composing a landmark L is defined with the following attributes:

- Center vector $\mathbf{R}_{rgb}$ in the color space,
- Geometrical central moments μ_{20}, μ_{11} and μ_{02} of the region,
- Squared patch $\mathbf{R}_{patch}$ to store a normalized version of the region's mask.

With a desired frequency, the current image frame is analyzed in order to characterize the landmark associated with the salient region defined by each unit U, according to the following steps:

1) Region mask determination:

Every pixel I in the image for which the unit responds is included into the salient region's mask.

$$mask_U(I) = 1 \Leftrightarrow \begin{cases} Mdist_{xy}(U,I) \leq MAXDISTXY \\ Edist_{RGB}(U,I) \leq MAXDISTRGB \end{cases} \tag{12}$$

In a second step, the region mask is expanded, possibly beyond the limits of the ellipse, by a growing process that includes those pixels connected to the mask that also satisfy the color constraint.

2) Landmark layout:

Once the region mask is obtained, we define the landmark layout as the convex hull of the mask. This allows to include regions of different colors into the landmark, and not only the single color region that was found as salient.

3) Extraction of relevant regions of the landmark:

The landmark layout is subject to a non-exhaustive segmentation process which tries to obtain its most relevant regions. The goal is to obtain a description of the landmark consisting in a small number of significant regions that cover

the most part of the area, excluding too small regions from the description. This is obtained by a process of color-based region growing initiated at successive randomly chosen seeds within the landmark layout. The process stops as soon as at least 80% of the landmark area has been segmented, or after a given number of seeds, determined in proportion to the region size, has been used.

During the growing process the region's color is updated as new pixels are included in the region's mask $\mathbf{R}_{mask}$:

$$\mathbf{d}_i = \mathbf{I}_i - \mathbf{R}_i \tag{13}$$

$$\mathbf{R}_i \leftarrow \mathbf{R}_i + \gamma \mathbf{d}_i \qquad \text{with } i \in \{r, g, b\} \tag{14}$$

$$\gamma = \frac{1}{R_{weight}+1} \tag{15}$$

$$R_{weight} \leftarrow R_{weight} + 1 \tag{16}$$

4) Region merging:

In order to make the segmentation more robust to initial seed selection, a post-process of merging is done, which joins those neighboring regions whose colors are similar enough. The result of merging two regions R^a and R^b is a new region R^c obtained as follows:

$$\mathbf{R}^c_{mask} = \mathbf{R}^a_{mask} \cup \mathbf{R}^b_{mask} \tag{17}$$

$$R^c_{weight} = R^a_{weight} + R^b_{weight}, \tag{18}$$

$$\mathbf{R}^c_i = \frac{R^a_{weight}\mathbf{R}^a_i + R^b_{weight}\mathbf{R}^b_i}{R^c_{weight}} \quad \text{with } i \in \{r, g, b\} \tag{19}$$

After this, regions representing a too small fraction of the landmark area are removed.

5) Region normalization:

Each region is normalized to a patch of 40x40 pixels to make its description invariant to changes in scale and moderate perspective deformations. For this, the geometric central moments of order 2 are computed, from which the equivalent ellipse axes are found providing a rotation angle to align the region, and scale factors for the x and y dimensions.

The geometrical moments are computed as:

$$m_{ij} = \sum_x \sum_y x^i y^j \mathbf{R}_{mask}(x, y) \tag{20}$$

$$\mu_{ij} = \sum_x \sum_y (x - \bar{x})^i (y - \bar{y})^j \mathbf{R}_{mask}(x, y) \tag{21}$$

$$\text{where} \qquad \bar{x} = \frac{m_{10}}{m_{00}}, \qquad \bar{y} = \frac{m_{01}}{m_{00}}$$

from which the equivalent ellipse orientation and axes are obtained:

$$\theta = \frac{1}{2} arctan\left(\frac{2\mu_{11}}{\mu_{20} - \mu_{02}}\right), \tag{22}$$

$$w = \sqrt{\frac{\mu_{20} + \mu_{02} + \Delta}{2m_{00}}} \quad \text{and} \quad h = \sqrt{\frac{\mu_{20} + \mu_{02} - \Delta}{2m_{00}}}, \tag{23}$$

$$\text{where} \quad \Delta = \sqrt{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2}.$$

The region mask normalization (translation, rotation and scaling) into a square patch $\mathbf{R}_{patch}$ is done in the following way:

$$\mathbf{R}_{patch}(x^{'}, y^{'}) = \mathbf{R}_{mask}(x, y) \tag{24}$$

$$(x^{'}, y^{'})^t = M * (x, y)^t + b \tag{25}$$

$$M = \begin{bmatrix} s_x cos(\theta) & s_x sin(\theta) \\ -s_y sin(\theta) & s_y cos(\theta) \end{bmatrix}, \quad b = \begin{bmatrix} t_x s_x cos(\theta) + t_y s_x sin(\theta) \\ -t_x s_y sin(\theta) + t_y s_y cos(\theta) \end{bmatrix} \tag{26}$$

$$\text{where:} \quad t_x = -\bar{x} \qquad t_y = -\bar{y} \qquad s_x = \frac{40}{2w} \qquad s_y = \frac{40}{2h}$$

Finally, a landmark description is stored as a set of regions, each defined by its characteristic color, its geometric moments and the normalized 40x40 region mask.

3.3 Landmark Identification

In order to identify previously encountered landmarks and update their description with the new view, each newly found landmark is compared against all landmarks currently in the landmark database. The comparison between two landmarks is done by trying to match the regions that form each landmark. Each region of the new landmark is compared with all regions of the stored landmark. Region comparison is performed according to two features: color and shape. The color match is simpler to test and is made first so as to act as a filter for the second test. Regions passing the color match filter are tested for shape similarity by computing the normalized cross-correlation NCC of the corresponding normalized masks:

$$NCC = \frac{\sum_x \sum_y \mathbf{R}^a_{patch}(x, y) \mathbf{R}^b_{patch}(x, y)}{\sqrt{\sum_x \sum_y (\mathbf{R}^a_{patch}(x, y))^2 \sum_x \sum_y (\mathbf{R}^b_{patch}(x, y))^2}} \tag{27}$$

If the shape similarity of two regions is higher than a threshold (defined as 85%), the correspondence between the two regions is added to a list of correspondences. This list may contain multiple matches for each landmark region, leaving the disambiguation of the correct matches for a later process of global coherence. A new landmark is identified with a landmark stored in the database if the percentage of matching regions is above 50%.

Non identified landmarks are added to the landmark database as new detected landmarks, so that they can be identified in later stages of the navigation process.

3.4 Experimental Results

The landmark detection and identification system was tested on real images taken during the navigation of a legged robot Lauron III (Figure 2). The Salience Detection algorithm was executed on the whole video sequence, while the processes of Landmark Characterization and Landmark Identification were only performed on the six isolated frames shown in Figure 3.

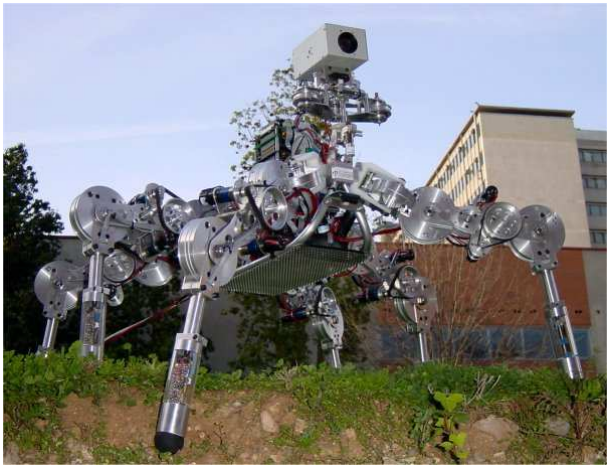

Fig. 2. The six-legged robot Lauron III used in the experiments.

Superimposed on the video images are the ellipses corresponding to the regions detected as salient by the Saliency Detection algorithm. Next to each image, the regions obtained for each landmark by the Landmark Characterization algorithm are shown. Arrows between frames indicate the landmark correspondences found by the Landmark Identification module. It can be observed that the blue bag and the two bright objects are consistently detected as relevant all the time the robot is approaching them. Other objects like the windows or the stairs are also found relevant most of the time they appear in the visual field. Other regions are eventually detected as salient, but not in a systematic way.

In the experiment, landmark correspondences are sought only between the landmarks of each frame with those of the previous one. As can be seen in the pictures, all detected landmarks that appear in two consecutive frames are correctly identified, except for the windows appearing in frames 1 and 2, which are only partially visible, and the bag in frame 5, which is largely out of view and can not be matched against the fully visible one of frame 4.

4 Conclusions and Future Work

We have presented a system for landmark detection, characterization and posterior identification, able to automatically select and track natural landmarks in arbitrary, non-structured environments. At the current stage, the system has shown the ability to reliably identify landmarks appearing in different views of the scene, taken at different stages of a navigation process performed by a legged robot. The new approach to detect and track salient regions provides an attention mechanism that serves as a filter of specific parts of the image, which can be further analyzed in detail to identify landmarks, thus allowing a real-time visual processing suitable for landmark-based navigation.

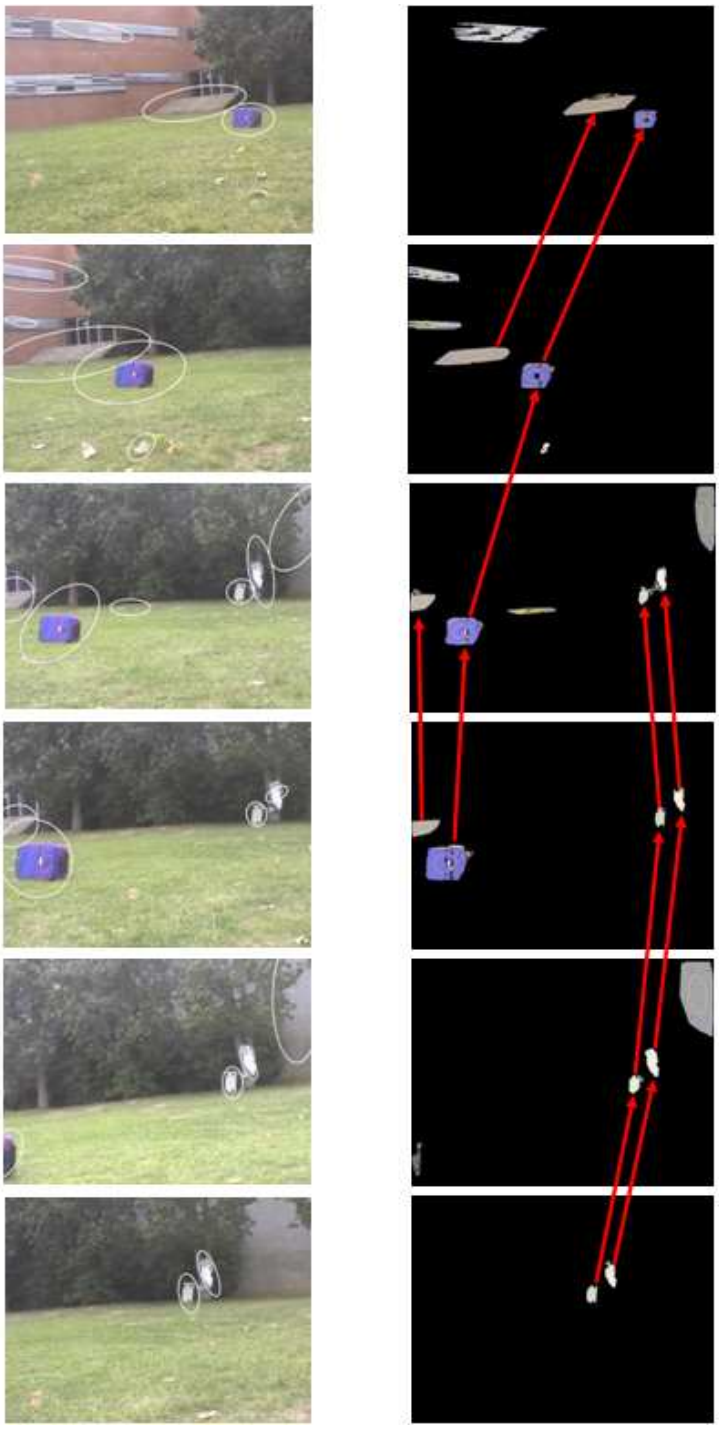

Fig. 3. Landmark detection and identification in an image sequence.

Further work is needed to make the landmark characterization more robust to large differences in the viewpoint and illumination conditions that may be expected during the travel of the robot in outdoor environments. As a future enhancement, a landmark could be represented by a collection of views taken form different locations at different stages of the navigation, thus coping with the problem of the appearance/disappearance of regions when changing the point of view. Another improvement concerns the algorithm for landmark matching: when the number of matched regions between two landmarks is not enough to assess a landmark identification, the missing regions could be actively sought in the landmark to confirm the correctness of the match.

Acknowledgements. This work has been supported by the Spanish *Ministerio de Ciencia y Tecnología* and FEDER under the project SIRVENT (DPI2003-05193-C02-01).

References

1. G. DeSouza and A. Kak: Vision for mobile robot navigation: a survey. *PAMI*, 24(2):237-267, 2002.
2. D. Lowe: Distinctive Image Features from Scale-Invariant Keypoints, *International Journal of Computer Vision*, vol. 60, 2004, pp 91-110.
3. P. Sala, R. Sim, A. Shokoufandeh, and S. Dickinson: Landmark Selection for Vision-Based Navigation, *IEEE Trans. on Robotics*, vol. 22, n. 2, 2006, pp. 334-349.
4. R. Murrieta-Cid, C. Parra and M. Devy: Visual navigation in natural environments: from range and color data to a landmark-based model. *Autonomous Robots*, vol. 13, n. 2, 2002, pp. 143-168.
5. E. Todt and C. Torras: Detecting Salient Cues Through Illumination-Invariant Color Ratios, *Robotics and Autonomous Systems*, vol. 48, n. 2-3, 2004, pp 111-130.
6. E. Celaya, J-L. Albarral, P. Jiménez, and C. Torras: Visually-Guided Robot Navigation: From Artificial To Natural Landmarks, 6th International Conference on Field and Service Robotics, Chamonix, France, July 2007.
7. D. Busquets, C. Sierra, and R. López de Màntaras: A multi-agent approach to qualitative landmark-based navigation, *Autonomous Robots*, vol. 15, 2003, pp 129-153.
8. H.-C. Nothdurft: Saliency from Feature Contrast: Additivity Across Dimensions, *Vision Research*, vol. 40, 2000, pp 1183-1201.
9. E. Todt and C. Torras: Detection of natural landmarks through multiscale opponent features, *in 15th International Conference on Pattern Recognition*, Barcelona, Spain, 2000, pp. 976-979.
10. L. Itti, C. Koch and E. Niebur: A Model of Saliency-based visual Attention for Rapid Scene Analysis, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, n. 11, 1998, pp 1254-1259.