Visually-Guided Robot Navigation: From Artificial To Natural Landmarks

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

Institut de Robòtica i Informàtica Industrial (CSIC-UPC), Llorens i Artigas 4-6, 08028 Barcelona, Spain. {celaya,albarral,jimenez,torras}@iri.upc.edu

Summary. Landmark-based navigation in unknown unstructured environments is far from solved. The bottleneck nowadays seems to be the fast detection of reliable visual references in the image stream as the robot moves. In our research, we have decoupled the navigation issues from this visual bottleneck, by first using artificial landmarks that could be easily detected and identified. Once we had a navigation system working, we developed a strategy to detect and track salient regions along image streams by just performing on-line pixel sampling. This strategy continuously updates the mean and covariances of the salient regions, as well as creates, deletes and merges regions according to the sample flow. Regions detected as salient can be considered as potential landmarks to be used in the navigation task.

1 Introduction

The robotics community has devoted many efforts to specific indoor navigation schemes for wheeled robots, often strongly dependent on structured visual features [1]. Nowadays, the demand for outdoor applications is increasing, and the research attention is shifting towards more general, though perhaps less accurate, navigation strategies that rely on unstructured visual cues.

Not only unstructured environments pose new challenges, but also the mobility capabilities of the robots themselves -outdoor robots usually use legs, tracks, or deformable frames- imply that concepts like what is actually an obstacle have to be reconsidered. Even the very goal of outdoor navigation is usually different from that of indoors. The typical goal in indoor navigation applications is to reach a desired place in a previously known environment, usually using a map, whereas in outdoor navigation, e.g., in a rescue task, the environment is often unknown and the goal must be specified on-line and redefined during the navigation task, based on the observations made from the positions already reached. Full teleoperation is not always possible, as shown by NASA's Mars Exploration Rovers project [2, 3] operating Spirit and Opportunity rovers in the surface of Mars, due to the time delay introduced by the travel of the control signal between the Earth and Mars. The possibility

of fully autonomous navigation may also be severely limited by the processing capabilities available to the rover's CPU.

Thus, in applications such as planetary exploration or rescue in catastrophic areas, 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. Updating the target as the robot advances, permits long-range journeys even with low-bandwidth and long-delay communications. Additionally, it would be desirable that these robots were able to find their way back autonomously without relying on any a priori knowledge of the environment. This so-called homing problem has been addressed in some works [4], where visual homing with bearings-based control is experimented in natural environments by using correlation of SIFT features [5] in images taken from different robot poses.

In our visually-guided navigation approach we have addressed these issues by relying on landmarks. Only few previous works have tackled a similar visually-guided landmark-based approach. In [6] a landmark-based environmental model is incrementally built using color and stereo range information. Then, the landmarks in the model are used for the robot localization and for long-range robot navigation, through the selection of different landmarks that form a sequence of sub-goals that the robot must successively reach. The system differs from ours in that, instead of being user-guided, it performs an automatic goal selection by searching for the landmark with the highest elevation peak among all detected landmarks in the scene.

The paper is structured as follows: Section 2 describes our approach to visually-guided navigation. In Section 3, we show the viability of the approach by testing it for indoor navigation using artificial landmarks. Section 4 introduces our approach to natural landmark detection, and some results are shown for real image sequences taken by a mobile robot in an outdoor setting. Finally, some conclusions and future work are pointed out.

2 Our approach to visually-guided navigation

A problem we had to face while designing our visually-guided navigation system was how to specify the navigation target. Map-based approaches, and those based on absolute coordinate systems like GPS, were discarded since we required our system to be useful also in completely unknown environments. Ideally, the user should be able to select any object in sight and mark it as navigation target. As the navigation proceeds, the user could select new targets, thus guiding the robot towards the desired location in a way similar to what a human explorer would do himself.

The problem with this approach is that, what can be a clearly identifiable object for the user, may be difficult to discern for an artificial vision system, so that the robot can be lost very soon. Our strategy to solve this problem consists in limiting the selection of possible targets to those that the robot is

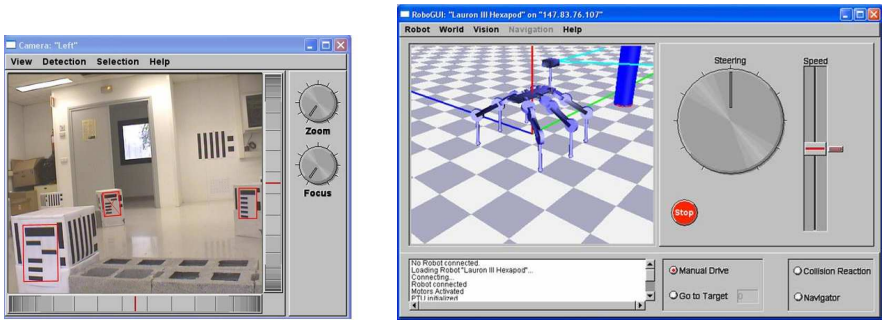

Fig. 1. Camera Window and Robot Control Interface.

able to identify with its vision system. Thus, it is the system that first shows to the user the available landmarks, and the user may select one of them as the target. To overcome the limitations that this approach could have in environments with sparse identifiable landmarks, it is possible for the user to directly drive the robot in the desired direction.

2.1 The Navigation Interface

Fig. 1 shows the two main windows of the navigation interface: The *Camera Window* and the *Robot Control Interface*. In the first, the user can see the images taken from the camera of the robot and control its gaze direction. Superimposed on the image, landmarks detected by the vision system are marked in red. The user may select one of these landmarks as the navigation target with a mouse click. Alternatively, the target can be selected using the Robot Control Interface, where landmarks are displayed in a simulated environment. In developing our platform for visually-guided navigation, our aim was to achieve a robot-independent control, so that the user could use it without being concerned about specific features of the robot that is performing the navigation task. The only assumptions we made about the robot are that it carries a camera, it is endowed with appropriate sensors to detect obstacles, and it is able of a short-term estimation of its own displacements (e.g. by odometry). For the tests of the system, two robots as different as a four wheeled Pioneer II and a six-legged Lauron III (Fig. 2) have been used.

The interface allows two control modes: *Manual Drive* and *Go to Target*. In the Manual Drive mode, the user directly controls the robot speed and direction with a slider and a driving wheel. Once a landmark has been selected as navigation target, the *Go to Target* mode can be triggered to let the robot reach the target in a completely autonomous way. Both modes can use the option *collision reaction*, which implements a high priority behavior that performs a pre-defined evading maneuver when an obstacle is detected. Obstacle detection is specific of each robot. In the case of the Pioneer, an activation of

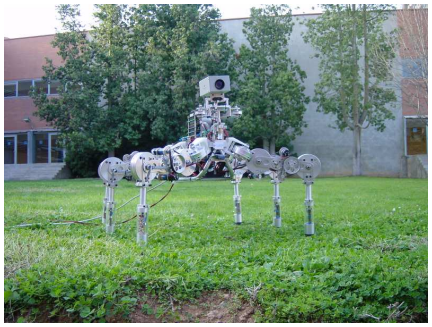

Fig. 2. The six-legged robot Lauron III used in the experiments.

the bumpers or a null advance of the robot in response of a movement command are interpreted as obstacles. In the legged robot, the activation above a threshold of the front infrared sensor, the collision with one leg at a given height, or the impossibility to find a stable support, are all considered as obstacles. Both robots also consider very close landmarks as obstacles.

In the *Go to Target* mode, by default, the robot simply heads directly towards the direction of the target. This simple reactive form of goal attraction, together with the collision reaction behavior, is enough to succeed in simple environments where path planning is not strictly necessary. For more complex environments, the *Navigator* option can be used to invoke a landmark-based navigation algorithm with map building and path planning capabilities. Our current implementation features the algorithm described in [7], but other landmark-based navigation algorithms could be used.

3 Navigating with artificial landmarks

A major problem in landmark-based navigation outdoors is a reliable identification of landmarks. In order to validate our visually-guided navigation approach, we isolated the landmark recognition problem and provisionally solved it by using artificial landmarks. For this, we designed a set of landmarks inspired by the work of Scharstein and Briggs in [8], who proposed the use of self-similar patterns, invariant to scaling, rotation, and viewing angle, whose detection indicates the presence of a landmark. A second pattern attached to the landmark composing a binary code served as identifier. This design has the drawback that most of the landmark area is used to indicate the presence of a landmark, but only a small area is devoted to the landmark code. This may prevent a correct identification of landmarks that are far away.

To improve on this, we modified the landmark design so that the code pattern is at the same time part of the landmark-presence indicator. The result is a set of landmarks like those marked in red in fig. 1, consisting in a header rectangle and five half-sized ones, whose relative positions, at the right or the

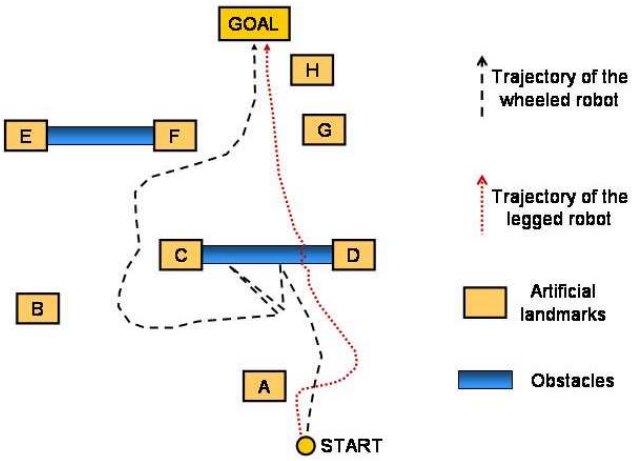

Fig. 3. Navigation experiment

left, allows the composition of $2^5 = 32$ different codes. Landmark detection tests performed with different cameras resulted in identification rates near 100% for non-occluded landmarks in the distance range from 60cm to almost 5m, provided the image is not too blurred due to fast camera movements. No false identifications of landmarks have been observed.

The navigation interface has been tested indoors using artificial landmarks with robots as different as a four wheeled Pioneer II and a six-legged Lauron III (Fig. 2). Users not familiar with the interface could easily drive the robot using the manual mode, no matter what robot was at work. On the other hand, the simple *Go to Target* mode, with the Navigator inactive, shows itself to be very intuitive. The collision reaction capability allows for a successful task completion even when the target is behind some obstacle. An unanticipated way to drive a robot was soon discovered: taking the selected landmark away causes the robot to track it, despite the fact that it is continuously moving or suddenly changes its position. With respect to the Navigator mode, it could be seen that the interaction between the navigation algorithm and the collision reaction behavior, naturally adapts the resulting trajectory to the actual capabilities of the robot performing the task, a feature that could be hard to implement in a more traditional map-based approach. To show this, we put the two robots in the same environment, and launched the navigator with the same target (Fig. 3). Since the legged robot has a much greater capability for obstacle crossing, what for the wheeled robot was an obstacle that caused the navigator to plan an alternative path, was considered just as an irregular patch of terrain by the legged robot, which followed its way walking over it with no further planning. Note that, for a map-based navigator to do the same, it should be provided with robot-specific maps, with different distributions of obstacles and free space.

4 Natural landmarks

Having shown that our visually-guided navigation approach was satisfactory, our next focus was on solving the problem of natural landmarks detection. For this, 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 [9]. This is also known as *opponency* [10]. Many variables can be used to define the saliency of a region, like color components, intensity, or feature orientation.Works like [11] 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.

4.1 Detecting Salient Regions: a random exploration approach

Instead of processing each incoming individual frame captured by the camera, the sequence of snapshots is treated as a continuously varying image. To this end, individual pixels are picked up at random from the current image and processed independently of whether the frame has already changed or not. In this way, a frame-rate and resolution-independent processing system is obtained, and the power of the computational resources only affects the number of pixels that can be processed per time step.

In the process, a number of so-called *units* are competing for the most relevant parts of the image, which we call Regions of Interest: each time a pixel is examined, the unit that best *responds* to it is selected and *updated*. By *responding* we mean to have a similar color while being in its proximity, and by *updating*, the process of adjusting the location and dimensions of the unit so as to improve its response to this pixel. In the case that no unit responds to a pixel, it may be considered as the seed of a new unit.

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 of the ellipse corresponds to the mean value of the pixels' positions, whereas its dimensions (the major and the minor axes) and orientation are given by the covariances. Each unit also has a spherical receptive field in the RGB color space centered at the mean color value and fixed radius, which is a parameter of the system. Units are implemented as data structures with the following fields:

- Center vector $\mathbf{U}_{xy}$ and covariance matrix $\mathbf{\Sigma}_{xy}$ in the image plane,
- Center vector $\mathbf{U}_{rgb}$ in the color space,
- Contrast C_U, a scalar that measures saliency, as explained below,

- Creation date, to record the time since 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.

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:

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

where $\mathbf{\Sigma}_{xy}$ is the covariance matrix of unit U, and

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

The Mahalanobis distance is used in the image space in order to take into account the shape of the spatial distribution. If both 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*.

Updating the winner unit

The center of the unit is approached to the coordinates of the pixel in the image plane as well as in color space:

$$U_i \leftarrow U_i + \gamma d_i, \tag{3}$$
$$\text{where } d_i = I_i - U_i, \text{whith } i \in \{x, y, r, g, b\} \text{ and } 0 < \gamma < 1$$

Covariances in image space determine the dimensions and orientation of the elliptic receptive field of the unit, and are updated according to:

$$\sigma_{ij} \leftarrow \sigma_{ij} + \gamma(d_i d_j - \sigma_{ij}), \qquad \text{where } i, j \in \{x, y\}. \tag{4}$$

Finally, the counter of updates is incremented by 1:

$$Updates_U \leftarrow Updates_U + 1 \tag{5}$$

Updating other units

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

$$C_U \leftarrow \alpha C_U + (1-\alpha)\sqrt{\sum_{i\in\{r,g,b\}} (I_i - U_i)^2}, \qquad 0 < \alpha < 1 \tag{6}$$

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 MAXDISTXY, 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:

$$S_U \leftarrow \beta S_U + (1-\beta), \qquad \text{with } 0 < \beta < 1. \tag{7}$$

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

$$S_U \leftarrow \beta S_U \tag{8}$$

Removing irrelevant units

To avoid an unlimited proliferation of units, their number is limited by a parameter of the system. To allow the creation of new units when the maximum number has been reached, the less useful ones must be removed. This is done in the following way: First, the unit with the lowest strength is sought. If its strength is below a given value, it is assumed that the region it represents is no longer there, and the unit is removed. Otherwise, a unit will only be replaced provided a contrast estimation of the unit to be created is above that of the lowest-contrast unit. The contrast estimation is given by the distance in color space of the input pixel to its spatially closest unit.

The new unit is initialized with a circular receptive field with center at the position of the input pixel and radius equal to its distance to the closest unit.

Merging of units

If two units respond to a given pixel, they are probably representing different parts of the same region and can be merged. The center and covariance values of the merged unit are computed as a weighted sum of those of the merging ones, with weights proportional to the respective area and strength, roughly corresponding to the "mass", or number of pixels each unit responds to:

$$Weight(U) = Area(U) S_U \tag{9}$$

The area is computed from the covariances as

$$Area(U) = \sqrt{((\sigma_{xx} + \sigma_{yy} + \Delta) \cdot (\sigma_{xx} + \sigma_{yy} - \Delta))} \tag{10}$$
$$\Delta = (\sigma_{xx} - \sigma_{yy})^2 + 4\sigma_{xy}^2$$

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 computed, this time using the respective values of $Updates_U$ as weights.

Output of the system

As for the output of the system, only the units that correspond to the most salient regions are returned as possible landmarks. 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 and output as landmarks.

4.2 Navigation using natural landmarks

Fig. 4 shows some snapshots of a preliminary test of the use of the salience detector for the guidance of a robot outdoors. The legged robot Lauron III is facing the situation shown in frame 1, in which the blue bag is chosen as initial navigation target. After some progress towards it (frame 2), two bright objects enter the field of view and are identified as salient landmarks by the system (frame 3). One of these objects is selected as a new target and the robot proceeds towards it letting the old target back (frames 4-6).

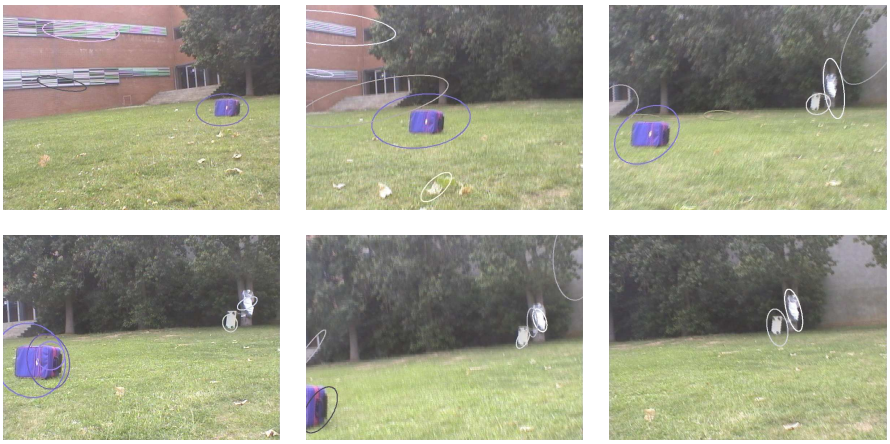

Fig. 4. Six snapshots of the navigation experiment with the robot Lauron III.

5 Conclusions

We have presented a working system for visually-guided navigation, which differs from others in the following features: (1) The interface is robot-independent even if the notion of obstacle varies from one robot to another, depending on their locomotion capabilities. Experiments have shown that the same navigation algorithm generates different trajectories for a legged and a

wheeled robot. (2) A partial teleoperation scheme has been adopted, in which the goal is selected by the user among a set of landmarks previously detected by the robot. (3) Navigation issues have been decoupled from vision ones by designing artificial landmarks that could be easily detected and identified. (4) Once we had a navigation system working, artificial landmarks were replaced by natural ones. A color-contrast saliency detection algorithm has been developed which relies on randomly sampling pixels from the continuous video stream acquired by an on-board camera. This algorithm has been tested on a legged robot in an outdoor scenario with promising results.

Acknowledgments: This work has been supported by the Spanish *Ministerio de Ciencia y Tecnología* and FEDER under the project SIRVENT (DPI2003-05193-C02-01), as well as the European Union under the project PACO-PLUS (FP6-2004-IST-4-27657). We thank the IIIA for making available the environment for the navigation experiments and performing the tests with the Pioneer 2, and also for providing the implementation of the navigator.

References

1. DeSouza G, Kak A (2002) Vision for mobile robot navigation: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, n. 2, pp. 237-267.
2. Leger P, Deen R, Bonitz R (2005) Remote Image Analysis for Mars Exploration Rover Mobility and Manipulation Operations, in IEEE Int. Conf. on Systems, Man and Cybernetics.
3. Maxwell S, Cooper B, Hartman F, Leger C and Wright J (2005) The Best of Both Worlds: Integrating Textual and Visual Command Interfaces for Mars Rover Operations, in IEEE Int. Conf. on Systems, Man and Cybernetics.
4. Kirigin I, Singh S (2005) Bearings based robot homing with robust landmark matching and limited horizon view. Technical Report CMU-RI-TR-05-02, Carnegie Mellon University.
5. Lowe D (2004) Distinctive Image Features from Scale-Invariant Keypoints, *International Journal of Computer Vision*, vol. 60, pp. 91-110.
6. Murrieta-Cid R, Parra C, Devy M (2002) Visual navigation in natural environments: from range and color data to a landmark-based model. *Autonomous Robots*, vol. 13, n. 2, pp. 143-168.
7. Busquets D, Sierra C, López de Mántaras R (2003) A multi-agent approach to qualitative landmark-based navigation, *Autonomous Robots*, vol. 15, pp. 129-153.
8. Scharstein D, Briggs AJ (2000) Real-time recognition of self-similar landmarks. *Image and Vision Computing*, vol. 19, pp. 763-772.
9. Nothdurft HC (2000) Saliency from Feature Contrast: Additivity Across Dimensions, *Vision Research*, vol. 40, pp. 1183-1201.
10. Todt E, Torras C (2000) Detection of natural landmarks through multiscale opponent features, *in 15th Int. Conf. on Pattern Recognition*, Barcelona, Spain, 2000, pp. 976-979.
11. Itti L, Koch C, Niebur E (1998) A Model of Saliency-based visual Attention for Rapid Scene Analysis, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, n. 11, pp. 1254-1259.