GeoBot: A High Level Visual Perception Architecture for Autonomous Robots

P.E. López-de-Teruel* A. Ruiz** L. Fernández*

*Dpto. de Ingeniería y Tecnología de Computadores
**Dpto. de Informática y Sistemas
Universidad de Murcia (Spain)

E-mail: {pedroe,lfmaimo}@ditec.um.es,{aruiz}@um.es

Abstract

This paper describes the software architecture of a mobile robot which is able to build in real time a structural interpretation of indoor environments using only visual and proprioceptive sensory information. Navigation is guided by this interpretation, improving on classical reactive approaches. We follow a predictive design criterion: the system must anticipate the consequences of its actions, showing predictive understanding of the scene. Specific solutions are given to all perception stages, from low level segment extraction to 3D scene reconstruction based on the current interpretation, including autocalibration of the camera-robot system. This paper focuses in the architecture that integrates all these elements into a high level perception system. A key point is the process of generation, tracking and confirmation of hypothesis which are maintained in a stable internal representation tuned with the agent movements. There is constant interaction between the bottom-up perceptive processes, guided by sensory stimuli, and the top-down ones, guided by the previously constructed models.

1. Introduction and previous work

Visual robot navigation is one of the most challenging computer vision problems. Real time image interpretation guides robot behavior and, at the same time, robot actions have effects in subsequent images frames. Even when restricted to indoor environments, visual navigation has been tackled using many different approaches. For example, many authors use natural or artificial *landmarks* that can be easily detected in the input images, assisting in robot self-localization [8]. Complete images themselves can also be used as landmarks, for example using appearance based methods, which trigger a preprogrammed response when a given location is recognized [19]. Recently, invariant features to many of the distortions of the imaging process (SIFT algorithm) have become very popular [17]. Other systems try to work in less restricted environments, and without any kind of *a priori* knowledge. For example, a direct navigation control loop can be closed to avoid obstacles using only the optical flow [3], or looking for free space in simple affine reconstructions of the scene [2]. This kind of approach, often termed as visual servoing, has received some criticisms due to the lack of scene *understanding* achieved by the robot [14]. Thus, the acquisition of a model of the environment as the robot operates (SLAM, for simultaneous localization and map building), has become a popular research topic. Some proposals include 2D [6] or 3D occupation grids [13] and simple accumulation of some kind of image features such as points [4], lines [20] or natural marks acquired during the navigation period [14]. The excellent reviews [1] and [18] (and references therein) discuss the main problems associated to visual navigation, such as egomotion estimation, obstacle and location recognition, stable vision, map building and spatio-temporal representations, among others.

In this paper we propose a perception-action architecture for autonomous robots operating in partially structured indoor environments. The robot elaborates an abstract model of the environment which is constantly updated. This model, computationally lightweight, allows a coherent navigational behavior under real time constraints and limited computation resources. The key is *interpretation*, in the sense of "matching sensor inputs to a certain model". In indoor scenes, walls, doors, and floor can be well modeled, in a first approximation, as planes. Our assumption is very simple: the world contains a set of horizontal and vertical planes in different orientations. Note that this is the only *a priori*, or *innate*, robot knowledge about the environment. It is planar at "large-scale", although the specific structure of the scene is unknown.

We have structured the paper as follows. Section 2 describes the planar model for structured environments.

Proceedings of the Fourth IEEE International Conference on Computer Vision Systems (ICVS 2006)
0-7695-2506-7/06 $20.00 © 2006 IEEE

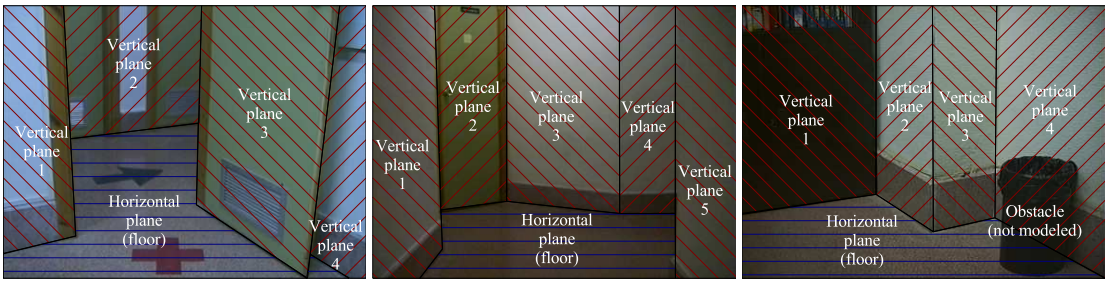


Figure 1. Planar models for interior scenes.

Section 3 briefly summarizes the low and medium level algorithmic components of the vision system. Section 4 describes the overall perceptual architecture of the robot, emphasizing the real time *interpretative-predictive-corroborative* processing cycle which is central in the proposal. Section 5 discusses the hardware and software implementation of the system, and presents some experimental results. Finally, we conclude in Section 6.

2. Planar representation of structured environments

Our perceptual architecture is admittedly representational. The raw visual input must be explained in terms of a suitable model. Indoor environments contain a number of nearly planar structures whose relative position is not completely arbitrary (walls and doors are orthogonal to the floor). These constraints notably simplify the geometry of the problem and will be conveniently exploited by the interpretation system. Fig. 1 shows a few examples of real images taken from the domain, with the relevant planes labeled.

This model is not fixed, or valid only for a predetermined building area. Rather, it is a *flexible model* of indoor environments. The only assumption is verticality of walls and doors with respect to the floor. Note also that the planar model is only intended to capture the scene *background*: Furniture, obstacles, moving persons and many other objects are obviously not modeled; see, for example, the bin in Fig. 1.c. As explained in Section 4, this is not a limitation unless excessive object clutter prevents detection of the relevant planes in the scene. Remarkably, although these "additional" objects are not explicitly contained in the model, they can still be approximately located in the 3D Euclidean scene reconstruction relative to the robot, allowing appropriate navigation actions such as obstacle avoidance.

In summary, the planar model is *scalable*: it is valid for general indoor environments (with vertical walls and doors) and the presence of other objects (if not very intrusive) does not interfere with the detection of the background planes.

3. Low and medium level vision algorithms

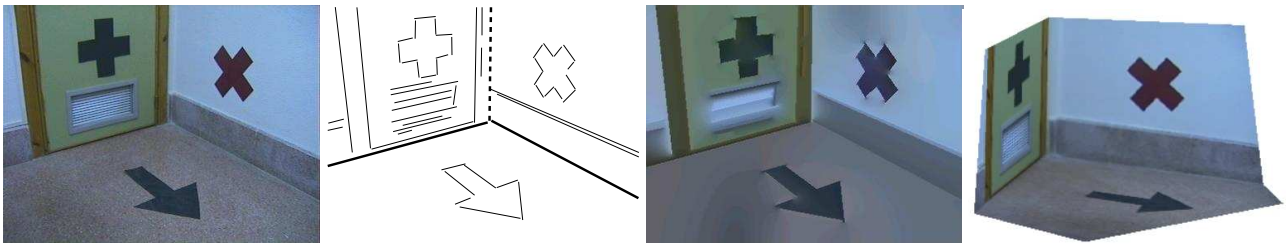
In this section we outline the low and medium level image processing techniques supporting the perception architecture. In addition to the classical references the reader can find more detailed descriptions in the PhD thesis [9].

At the lowest level, the system efficiently extracts straight line segments augmented with robust local color information [12]. This geometric primitive is specially adequate for partially structured scenes (Figs. 2.a-b). The relevant geometric and photometric properties of images are concisely captured (Fig. 2.c), while effectively reducing the high data bandwith of the input image sequence. This is crucial to comply with the real time constraints demanded by navigation.

To infer Euclidean properties of space the system must calibrate its camera [5]. A collection of autocalibration procedures have been designed to determine the relevant intrinsic and extrinsic camera parameters from the previously extracted segments [10, 9]. These techniques take full advantage of the odometric information supplied by robot, which significantly simplifies the calibration problem. The camera projection matrix $\mathtt{P} = \mathtt{KR}[\mathtt{I}| - \boldsymbol{C}]$ is explicitly factorized in terms of the position $\boldsymbol{C}$ and rotation $\mathtt{R}$ of the camera in the robot coordinate system, and the intrinsic parameters matrix $\mathtt{K}$, essentially characterized by the focal length.

A crucial module is devoted to monocular 3D scene reconstruction by exploiting the planar model constraints. Let us suppose, for the moment, that we have correctly *interpreted* the image segments (as explained in the next section), and therefore we know the intersections of the relevant planes in the scene (marked bold in Fig. 2.b) The input image can then be divided into $N+1$ zones, corresponding to N vertical planes plus the ground plane. The most important segment in each vertical plane is the intersection with the ground (i.e., the *base*, marked solid bold in Fig. 2.b). It can be shown [11, 9] that the 2D image $\boldsymbol{p}'$ of any point in a vertical plane can be back-projected to its true 3D position $\boldsymbol{P}$ in the robot frame just from the line $\boldsymbol{l}'$ containing the *base* segment and the camera matrix $\mathtt{P}$. Point reconstruction

Proceedings of the Fourth IEEE International Conference on Computer Vision Systems (ICVS 2006)
0-7695-2506-7/06 $20.00 © 2006 IEEE



(a) (b) (c) (d)

Figure 2. Illustration of the low and medium level processes: (a) Typical input image. (b) Extracted segments (local color is not displayed). (c) Recovered image using just the geometric and chromatic information contained in the segments, which take only about 1% of the storage required by the original uncompressed image. (d) 3D textured reconstruction of the scene using the planar model (see text).

can be concisely expressed as

$$\boldsymbol{P} = \mathtt{V}(\boldsymbol{l}') \cdot \boldsymbol{p}' \tag{1}$$

where $\mathtt{V}(\boldsymbol{l}')$ denotes the 4×3 matrix defined by

$$\mathtt{V}(\boldsymbol{l}') = \mathtt{C}(\tilde{\mathtt{P}}^{\top}\boldsymbol{l}') \cdot (\mathtt{P} \cdot \mathtt{C}(\tilde{\mathtt{P}}^{\top}\boldsymbol{l}'))^{-1} \tag{2}$$

in terms of

$$\mathtt{C}(\boldsymbol{l}) = \mathtt{C}(a,b,c) = \begin{pmatrix} b^2 & -ab & -ac \\ -ab & a^2 & -bc \\ a\sqrt{a^2+b^2} & b\sqrt{a^2+b^2} & c\sqrt{a^2+b^2} \\ 0 & 0 & a^2+b^2 \end{pmatrix} \tag{3}$$

and $\tilde{\mathtt{P}}$, which is the ground to image homography (the camera matrix $\mathtt{P}$ with its third column dropped). On the other hand, points in the ground plane are reconstructed simply using $\tilde{\mathtt{P}}^{-1}$. The process is illustrated in Fig. 2.d.

This projective geometry result is extensively used for hypothesis generation and scene interpretation, as described in the next section.

4. Perceptual architecture

Fig. 3 summarizes the proposed perception-action architecture. The main information structures appear in boxes and the corresponding data flows are represented by labeled arrows. The boxes are arranged in a semi-tabular form. Columns denote the reference frame: 2D *image coordinates* for points, segments, contours and so on, 3D *space robocentric* coordinates for constructed structures such as 3D segments or planes, and 2D *robocentric control* coordinates for movement control orders. Rows denote different levels in the perceptual hierarchy (see [16]), managing the following types of information:

- **Sensory level:** Raw input images, robot odometry (self-localization information), and camera calibration parameters.
- **Primitive level:** Intermediate perceptual structures such as the extracted colored segments and the 3D reconstructions described in Section 3.
- **Aggregation level:** Geometric adjacency and chromatic compatibility is used to interpret some segments as intersections of planes and for creation of tentative structures (e.g. closed contours) that can be subsequently interpreted as different kind of objects.
- **Interpretative level:** Planes are used to create higher level models of the scene (corridors, corners, walls, etc.) that we call *schemaps*. Contours can also be interpreted as signals and obstacles. All these structures are used to elaborate a perceptual categorization of the local environment which governs robot navigation. At the same time, this local interpretation is incorporated, using odometry, into a global map containing both *topological* information (each situation is labeled as a corridor, a room, etc.) and precise *metric* information (objects are explicitly situated in real 3D space coordinates).

The processing scheme is based on a repeated cycle in which bottom-up, data-driven heuristics create tentative, increasingly complex structures that are corroborated or rejected when the robot moves, depending on reprojection quality on new image frames. As explained below, odometric information is used to *predict* where to look for segments supporting the current interpretation.

The bottom-up processes try to find some relations among primitives. For example, segments with nearby extremes and compatible color are joined to form closed contours (possibly corresponding to signals or obstacles). Long

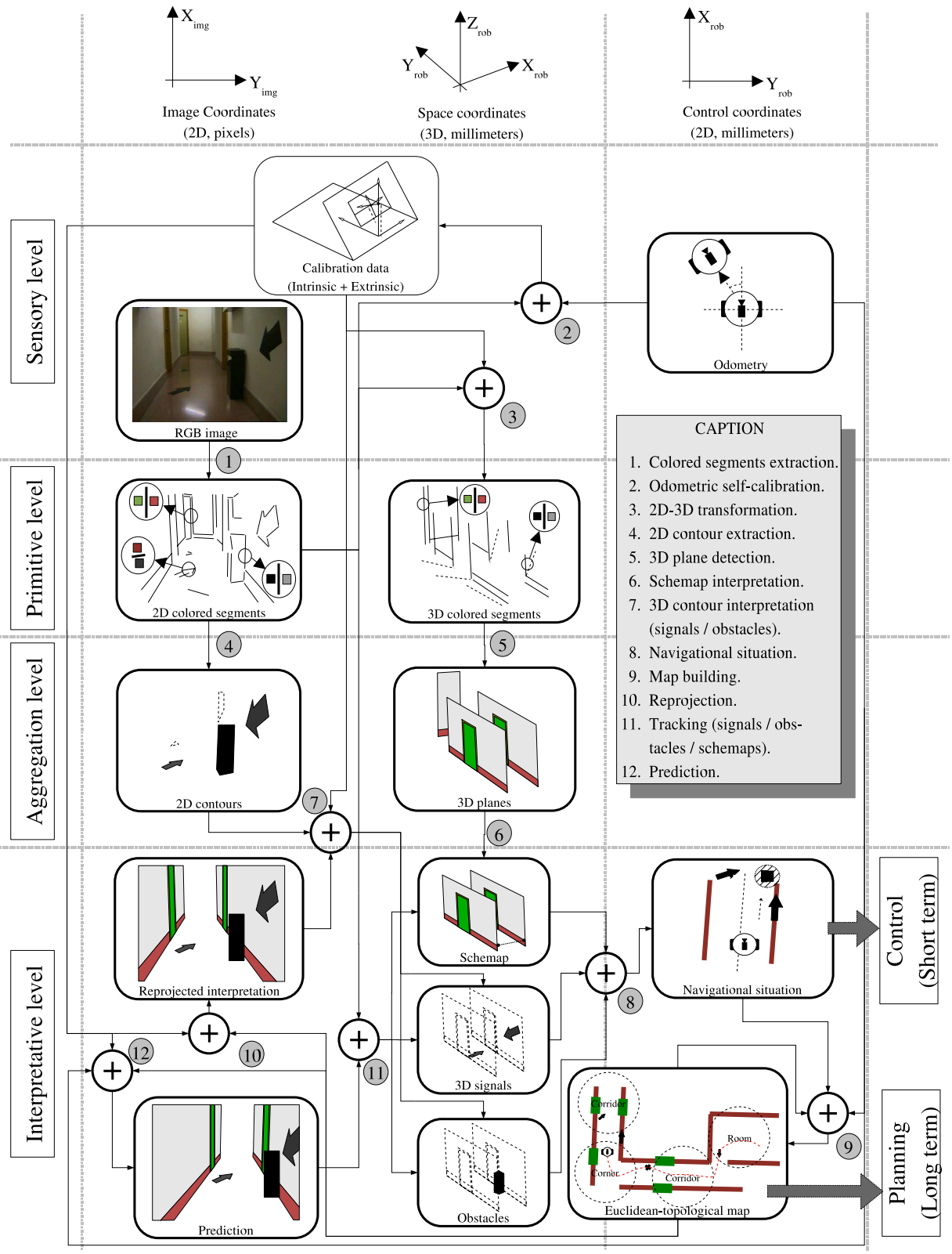


Figure 3. High level perception architecture.

Proceedings of the Fourth IEEE International Conference on Computer Vision Systems (ICVS 2006)
0-7695-2506-7/06 $20.00 © 2006 IEEE

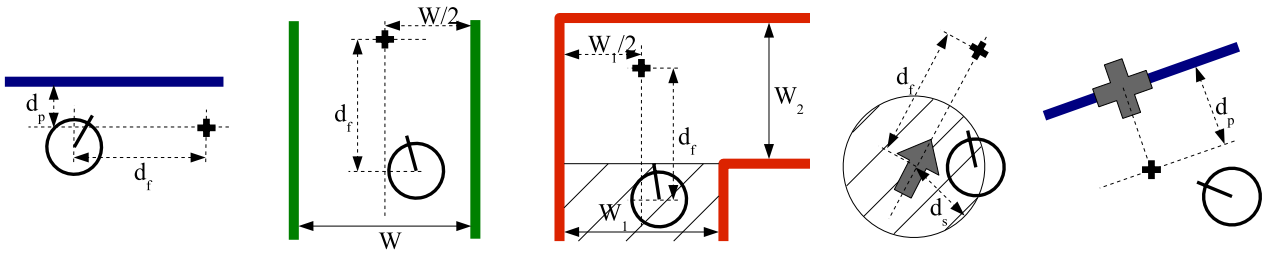


Figure 4. Different examples of *schemaps* and signals recognized by the robot. Left to right: Wall to the left, corridor, corner to the right, arrow on the floor, cross on a wall.

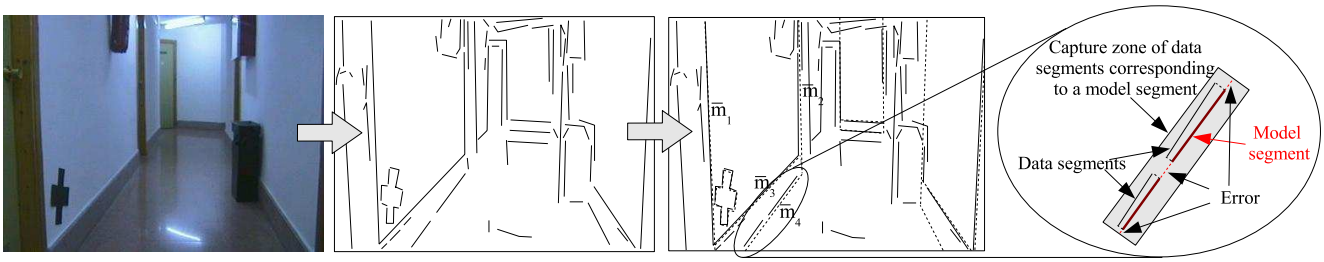


Figure 5. Left to right: original image, segment extraction, reprojection of current interpretation (corrected by odometry), and computation of reprojection error.

segments whose lower color is compatible with that of the floor are candidates for the *base* of a wall. Nearby vertical segments can then be found using $\tilde{\mathtt{P}}$ and $\mathtt{V}(\boldsymbol{l}')$. This way, initial hypotheses of vertical planes are generated for the reconstruction procedure described in Section 3. The posterior aggregation of vertical planes into *schemaps* (Fig. 4) is also considered a bottom-up procedure.

Depending on the situation of the robot with respect to the *schemap*, a short term navigation path is planned (to avoid a wall, navigate along a corridor, turn a corner, follow an arrow, etc.). This is accomplished by continuous updating of a *control point* (a black cross in the figures) which specifies the instantaneous intention of movement. Motor control orders are given to the robot to follow this target point, causing an observable behavior which is consistent with the current interpretation of the environment.

Unfortunately, it is very difficult to build complete perceptual structures directly from the low level image primitives. For example, segment fragmentation complicates the detection of the limits of a plane, and a full combinatorial search is computationally prohibitive. Obviously, the bottom-up processes by themselves cannot meet the requirements imposed by real time navigation. Robot responses would be unacceptably slow and the inevitable interpretation failures would seriously degrade the stability of behavior. Complementary top-down processes, guided by previously detected structures, are essential. Conjectured substructures will be then confirmed or discarded depending on their stable presence in subsequent frames, as the robot moves in the environment. This kind of interaction between bottom-up and top-down processes is one of the key aspects of the architecture.

We opt for efficient hypothesis generation by establishing very demanding preconditions (for example, working only on a well conditioned set of input primitives). Due to world continuity, after a few attempts the structure will be eventually captured in a well conditioned frame. From then on, the top-down processes will try to constantly corroborate or reject the detected structure, predicting and tracking its position as the robot moves. The process is illustrated in Fig. 5. Prediction failures discard tentative structures, protecting against possible bottom-up misinterpretations.

If, on the contrary, the prediction finds support in the segments extracted from subsequent frames, the structure is considered stable and is incorporated into the currently confirmed interpretation. Observe that bottom-up "constructor" processes are no longer needed for segments corresponding to formerly detected structures: they need only be tried on segments not yet "explained", thus alleviating the computational load. We call this top-down and bottom-up interaction *interpretative hysteresis*, as there is an initial resistance to detect a new structure, but once detected and corroborated, the structure is easily tracked and updated at a very low computational cost.

The computation of the reprojection error is illustrated in Fig. 5. There are three vertical planes, each formed

Proceedings of the Fourth IEEE International Conference on Computer Vision Systems (ICVS 2006)
0-7695-2506-7/06 $20.00 © 2006 IEEE

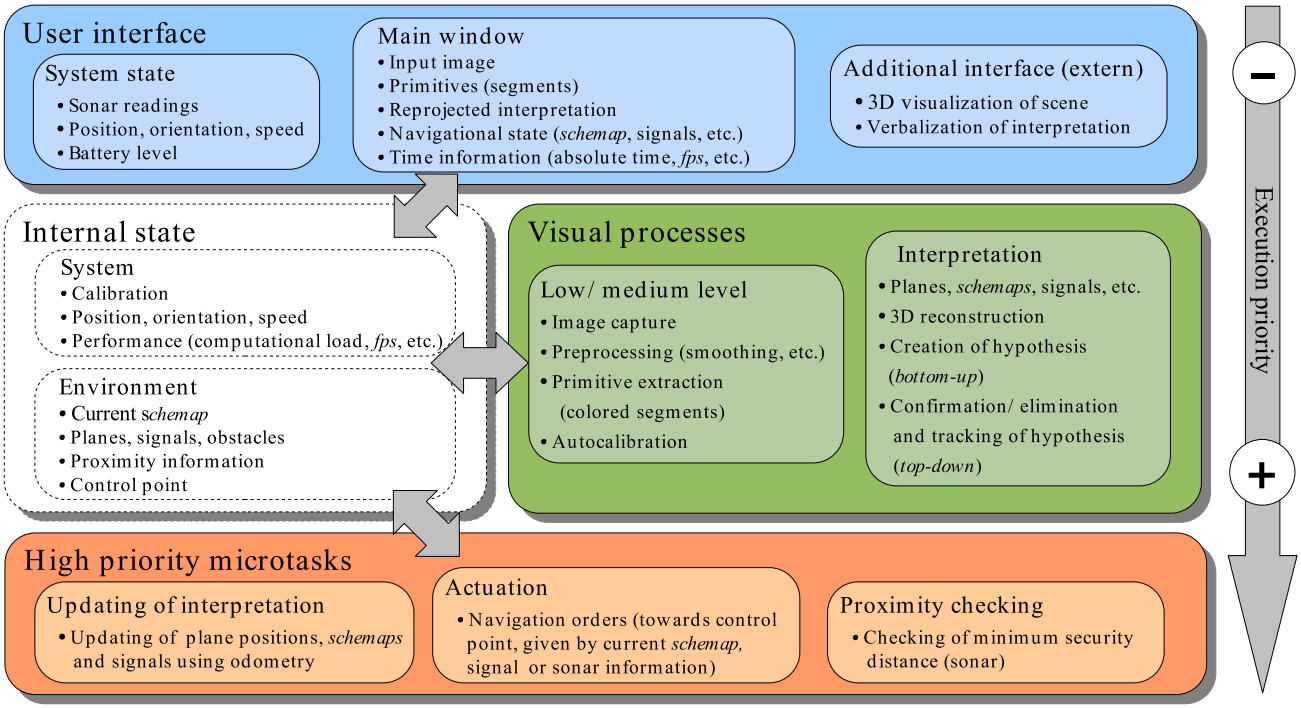


Figure 6. Software architecture of the system.

by four segments (dashed lines). For every reprojected model segment $\overline{\mathbf{m}}_j$ we define a *capture zone* with a predetermined width (see detail in Fig. 5). Then we collect the set $Match(\overline{\mathbf{m}}_j)$ of candidate input segments with extremes in the capture zone and compatible local color. The following expression gives us the *omission error* for each $Model = \{\overline{\mathbf{m}}_j\}$ and set of input segments $Data = \{\overline{\mathbf{d}}_i\}$:

$$E_{om}(Model, Data) = 1 - \frac{\sum_{\overline{\mathbf{m}}_j \in Model} \sum_{\overline{\mathbf{d}}_i \in Match(\overline{\mathbf{m}}_j) \subseteq Data} \mathcal{P}(\overline{\mathbf{d}}_i)}{\sum_{\overline{\mathbf{m}}_j \in Model} \mathcal{L}(\overline{\mathbf{m}}_j)} \quad (4)$$

where $\mathcal{L}(\overline{\mathbf{m}}_j)$ is the length of the model segment $\overline{\mathbf{m}}_j$ reprojected in the image and $\mathcal{P}(\overline{\mathbf{d}}_i)$ is the length of the orthogonal projection of each data segment $\overline{\mathbf{d}}_i$ onto the corresponding $\overline{\mathbf{m}}_j$ (overlapping zones are added up only once). Thus, $E_{om} \in [0, 1]$ is the proportion of the reprojected model not "covered" by input data. If E_{om} is under a certain tolerance threshold, the model is confirmed and maintained in the internal state of the agent. Otherwise, after a reasonable interval of time without adequate support (to be resistant against temporary low-level extraction failures and occlusions produced by moving objects), the structure is rejected. In coherency with the above *interpretative hysteresis* principle, E_{om} does not penalize fragmented segments, as long as they cover the majority of the model; the creation of new hypotheses needs non-fragmented segments, but, once detected, tracking is easy and lightweight.

5. Implementation and experimental results

The proposed architecture has been implemented on a Pioneer 2 DX robot, equipped with an onboard laptop running Linux and a 45° field of view camera (Fig. 7.a). The robot runs a small microkernel OS which executes low-level tasks such as movement control, sonar and odometry readings, and communication with the external computer. Robot responsiveness requires a fast processing loop, so the most expensive computer vision routines take place on the laptop, and are built on top of the optimized *Intel IPP libraries* [7].

Fig. 6 summarizes the software organization of the system. Processing modules are arranged by priority of execution. The highest priority corresponds to low-level sensors, control and security related tasks (emergency sonar checks for nearby objects, constant update of the current interpretation using odometry, etc.). These processes directly interact with the underlying robot OS, and are executed at a minimum rate of 10 Hz in the current implementation. The high level visual interpretation process is executed with lower priority, using the available CPU time. Finally, the user interface routines are invoked only sporadically. All modules have concurrent access to the *internal state* of the

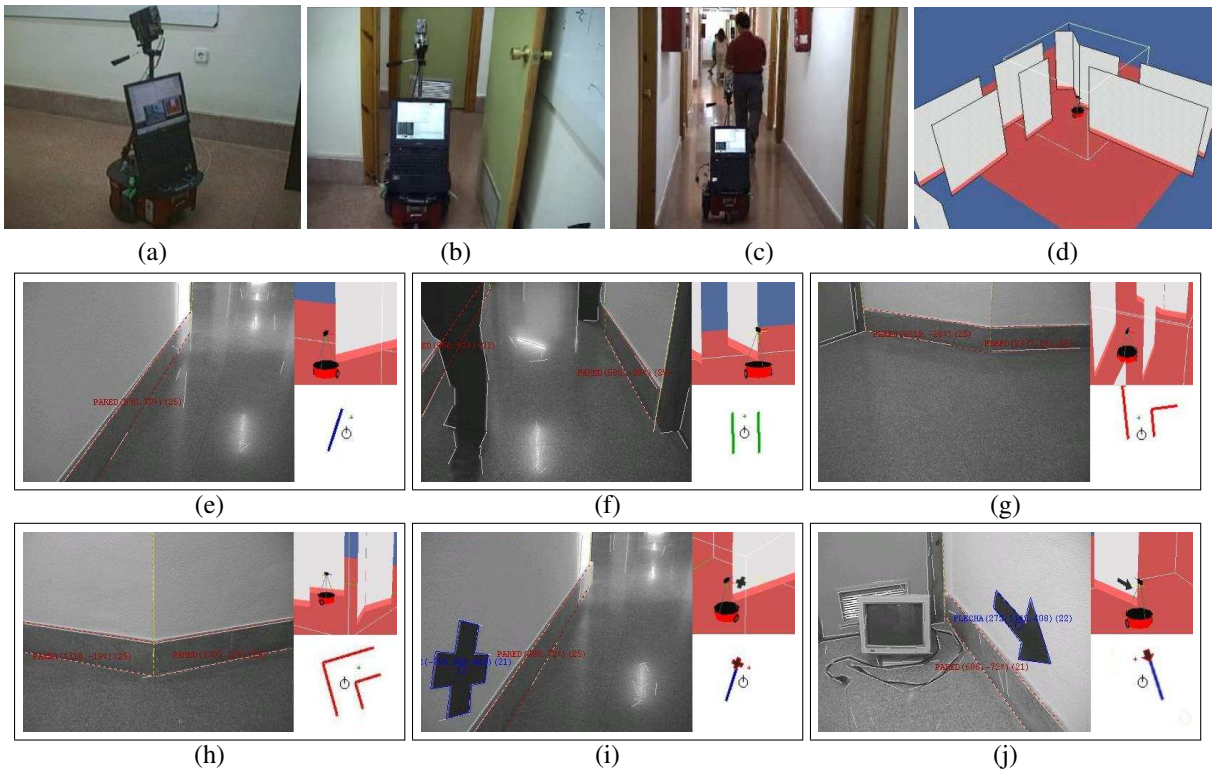


Figure 7. Illustration of a real navigation experiment.

robot, the main communication channel among the different processes. It contains information on both the internal system (position, speed, calibration parameters, CPU load, etc.) and the external environment (*schemaps*, signals, topological map, etc.). Concurrent access to shared data uses adequate synchronization mechanisms.

Some external views of GeoBot operation are shown in Figs. 7.a-c. The approximate size and structure of a typical building area can be appreciated in the 3D reconstruction of the environment (Fig. 7.d) performed by the robot itself. In contrast to classical grid maps [6, 13], the storage and computational load needed to update the map scale well with the size of the environment, as the involved information is limited to a small set of geometric primitives. Figs. 7.e-j show the high-level interpretation of several domain situations, including different *schemaps* and signals whose detection and tracking ultimately guide the navigation. We include the current interpretation reprojected on the original image (left), and the local 3D model and movement target (right). Despite projective deformations, signals on the floor or in any vertical plane can be easily interpreted using the monocular reconstruction procedure described in Sec. 3.

In any case, the system can be better judged by watching the dynamic behavior of the robot. A collection of sample videos is available at the GeoBot homepage [15]. The available processing power allows for a 5 fps average rate (depending on scene complexity) from 288×384 input images, which suffices for real time robot navigation at 20 cm/s. Using a modern laptop (e.g. 2$\sim$3 GHz CPU) the frame rate can be significantly increased. Note that the robot performs smooth survey navigation using pure visual information during several minutes, without crashing with walls, static objects or moving persons. In normal conditions, the robot switches to emergency sonar control mode (due to temporary failure of visual interpretation) only about 1% of total navigation time.

6. Conclusions and future work

In this paper we have described a complete computer vision system for indoor robot navigation. The proposed architecture is grounded on a number of efficient low and medium level techniques well adapted to the application domain, including colored segment extraction, odometric autocalibration and monocular 3D reconstruction based on the interpretation of the scene. These elements are integrated into a high level perception scheme guided by dual top-down and bottom-up *cognitive pressures*. The load is dynamically balanced among harder processes which create new interpretations from scratch and lighter ones which keep updated the detected structures. The experiments performed with our robot in real indoor environments support

the viability of the proposal.

Further work is possible in multiple directions. For example, a more formal approach to uncertainty in 3D reconstruction and incremental map building can be performed using Kalman filtering [20, 4] or any other kind of Bayesian updating [6]. Extending the interpretative paradigm to include more complex structures is another interesting possibility. Finally, the integration of other well studied approaches, such as those mentioned in the introduction, into the proposed architecture opens up a number of promising research lines.

Acknowledgements

This work has been supported by Spanish MCyT grants TIC-2003-08154-C06-03 and DPI-2001-0469-C03-01. The authors would also like to thank the reviewers for their careful reading and useful suggestions.

References

[1] Y. Aloimonos, editor. *Visual Navigation: From Biological Systems to Unmanned Ground Vehicles*. Lawrence Erlbaum Associates, New Jersey (USA), 1997.

[2] P. Beardsley, I. Reid, A. Zisserman, and D. Murray. Active visual navigation using non-metric structure. In 5^{th} *IEEE Int. Conf. on Computer Vision*, Cambridge (USA), 1995.

[3] T. Camus, D. Coombs, M. Herman, and T. Hong. Real-time single-workstation obstacle avoidance using only wide-field of view divergence. *Videre*, 1(3), 1999.

[4] A. Davison. Real-time simultaneous localisation and mapping with a single camera. In 9^{th} *IEEE Int. Conf. on Computer Vision*, Nice (France), 2003.

[5] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge (UK), 2000.

[6] A. Howard and L. Kitchen. Fast visual mapping for mobile robot navigation. In *IEEE Int. Conf. on Intelligent Processing Systems*, Beijing (China), 1997.

[7] Intel Corporation. The Intel integrated performance primitives (IPP) software library homepage, 2005. Available in `http://www.intel.com/software/products/ipp/`.

[8] W. Lee, K. Roh, and I. Kweon. Self-localization of a mobile robot without camera calibration using projective invariants. *Patt. Recogn. Letters*, 21, 2000.

[9] P. López-de-Teruel. *Una Arquitectura Eficiente de Percepción de Alto Nivel: Navegación Visual en Tiempo Real para Robots Autónomos en Entornos Estructurados (in Spanish)*. PhD thesis, Computer Engineering Departament, University of Murcia, Spain, 2003. Available in `http://ditec.um.es/~pedroe/documentos/Tesis.pdf`.

[10] P. López de Teruel and A. Ruiz. Closed form self-calibration from minimal visual information and odometry, 2005. (*Submitted*).

[11] P. López de Teruel, A. Ruiz, and L. Fernández. Efficient monocular 3D reconstruction for visual navigation in structured environments, 2005. (*Submitted*).

[12] P. López de Teruel, A. Ruiz, G. García, and J. García. Real-time extraction of colored segments for robot visual navigation. In 3^{rd} *Int. Conf. on Computer Vision Systems*, Graz (Austria), 2003.

[13] H. Moravec. Robot spatial perception by stereoscopic vision and 3D evidence grids. Technical Report CMU-RI-TR-96-34, Carnegie Mellon University, September 1996.

[14] E. Riseman, A. Hanson, J. Beveridge, R. Kumar, and H. Sawhney. Landmark-based navigation and the acquisition of environmental models. 1997. In [1].

[15] A. Ruiz and P. López de Teruel. GeoBot project homepage. University of Murcia (Spain), 2003. Available in `http://dis.um.es/~alberto/geobot/geobot.html`.

[16] S. Sarkar and K. Boyer. Perceptual organization in computer vision: A review and a proposal for a classification structure. *Syst., Man and Cybernetics*, 23(2), 1993.

[17] S. Se, D. Lowe, and J. Little. Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks. *Robotics Research*, 21(8), 2002.

[18] S. Thrun. Robotic mapping: A survey. Morgan Kaufmann, 2002. In G. Lakemeyer and B. Nebel, editors, *Exploring Artificial Intelligence in the New Millenium*.

[19] J. Weng, S. Chen, and T. Huang. Visual navigation using fast content-based retrieval. 1997. In [1].

[20] Z. Zhang and O. Faugeras. A 3D world model builder with a mobile robot. *International Journal of Robotics Research*, 11(4):269–285, 1992.