

Selecting Landmarks for a Visual Based Navigation Task

Swain-Oropeza Ricardo Burschka Darius Kriegman David Hager Greg Knappek Markus*

Tec de Monterrey CEM	Johns Hopkins University	University of Illinois	Technical University of Munich
Computer Science	Computer Science	Computer Science	Computer Science
Edo. de México	Baltimore	Urbana-Champaign	Munich
México	USA	USA	Germany

Abstract. The aim and scope of this paper is to present an overview of research made until now by our collaborative group in the field of sensor-based robots. Our mission in an unknown environment is select natural landmarks and then, its use in a visual-based navigation task (in this case everything has been tested in structured, but unknown environments). Many approaches of visual servoing and mobile robot navigation are based on artificial landmarks in images. Here we present a method that select natural landmarks (perceptually salient and visually distinctive), then, using this landmarks, our robot moves and use them to repeat the task many times(a visual-based navigation task). Experimental results will be presented.

1 Introduction

Sensor-based autonomous robotics has a long and rich history, however, only very recently have robust, truly autonomous sensor-based robots become a realistic near term possibility, largely due to technological advances, including the development of fast computing, large inexpensive computer memories, sophisticated sensors, and low-power motors. With this explosion in technological capabilities, a number of problems that were previously considered intractable have become topics of active research. Among these are problems that certain to active exploration by robots in previously unknown environments.

Almost all approaches on visual servoing are based on tracking feature points in an image sequence that correspond to the projection of viewpoint independent features of the 3-D scene or object [1, 8, 11, 19]. Similarly, numerous approaches to vision-based mobile robot navigation recognize and possibly track landmarks [4, 12, 14, 17, 21]. Except within specific applications, most visual servoing implementations in mobile robots have either used a catalogue of model/application specific landmarks or relied on a person to initialize tracking. In most mobile robotics implementations, landmark recognition system is provided with a catalogue of domain-specific recognizable landmarks (e.g. lane boundaries, ceiling light, bar codes, door edges, etc.) [10, 19]. In this days, it has become possible to track at frame rates a modest number of features (12 for example) on conventional personal computers using XVision [5]. Yet in a scene or for an object of interest, there may be hundreds or thousands of feature points which could serve as landmarks. For real-time robot control using current methods, only a fraction of the possible features can be considered and tracked. One way to select these promised natural landmarks (which are very *trackable* -salient- and readily recognized -distinctive-) is using our method described in [9]. For a visual-based navigation task using natural landmarks, once that we have the selected landmarks, we need to initialize our system and then, perform our task. Our goal is not only perform this task but also achieved many times.

This paper is organized as follows: in next section § 2 we describe the problem and the way we choose to solve it in order to perform a visual-based navigation task using natural landmarks, § 2.1 explain our method to select landmarks, § 2.2 shows the visual-based navigation (visual servoing) approach use it, and finally § 3 presents some results obtained using this approach.

* Email: rswain@campus.cem.itesm.mx, burschka@cs.jhu.edu, kriegman@uiuc.edu, hager@cs.jhu.edu,
 mknappek@brain.nefo.med.uni-muenchen.de

2 Methodology

This paper outlines a set of problems associated with constructing a vision-based navigation system suitable for structured and unstructured environments. The system utilizes natural landmark selection in order to initialize our system and then using visual-based navigation techniques, our mobile robot could repeat a path several times.

Recent advances in visual servoing theory and practice now make it possible to accurately and robustly position a mobile robot relative to a target. In this case, both vision and control algorithms are simple, but they must be initialized on task-relevant features in order to be applied. Our natural landmark selection gives a good accurate for this process.

2.1 Landmark selection

Many methods for tracking have been developed including corner tracking, line tracking, region tracking, etc. In general, trackers continually estimate some parameter vector representing some attributes of the tracked object (e.g. image location, scale, lighting, etc.) which are presumed to be varying continuously. One class of trackers that is particularly useful for robot navigation provides over time the image location of the projection of a local (small) region or point of a 3-D scene. Tracking a modest number of such features can be used to localize the robot, navigate using visual servoing, or recognize a place. Typically, the local region is represented by a template. When tracking, a region of an image is searched for the location which minimizes the sum of squared differences (SSD) between the template and the image intensities about the location. Due to 3-D viewpoint changes however, the image pattern will differ from the template, and this is sometimes modeled as an affine image warp [5].

Here, our goal is to select distinctive landmarks from one image which can be readily recognized in a second image acquired from a different viewpoint, see for more details [9]. Doing this, our system guarantee that landmark are easily trackable during a sequence.

Our landmark selection method should be useful for visual servoing and should be the projection of viewpoint independent features (e.g., points, corners, junctions, etc).

Let the irradiance (intensity) across the image plane be denoted by $I(x, y)$ where x and y are the image coordinates. Since we are interested in tracking the projection of point-like features, we can characterize $I(x, y)$ locally about a point (x_0, y_0) by its differential structure. In particular, consider the vector of partial derivatives up k-th order which is known as the k-jet; for example, the 2-jet of $I(x, y)$ is given by:

$$\mathcal{F}(x, y) = \begin{bmatrix} I \\ I_x \\ I_y \\ I_{xx} \\ I_{xy} \\ I_{yy} \end{bmatrix}$$

The k-jet or some function of the k-jet can serve as a representation of a landmark. We now summarize the basic landmark selection and recognition method which directly follows elements of the recognition method of Schmid and Mohr [15].

To select the landmarks:

1. A detector is applied to the entire image to select potential landmarks which should be readily tracked (salient).
2. The potential landmarks are characterized by a feature vector derived from the k-jet.
3. The potential landmarks are ordered by distinctiveness, and the most distinctive ones are retained.

Similarly, the landmarks are recognized in a second image:

1. The same detector is applied to the image, but with lower thresholds, to identify candidate locations of landmarks.
2. Each candidate is again characterized by a feature vector computed from the K-jet.
3. Each selected landmark is recognized independently by nearest neighbor classification using a Mahalanobis distance.
4. Lower error rates is possible by using geometric constraints of multiple matches.

The k-jet clearly depends on the location and orientation of the camera, but since we consider local surface descriptions and images, we develop a feature representation which is invariant to linear image transformation. If the camera motion is constrained or if elements of the camera motion can be directly measured, then we may only be interested in invariance to certain subgroups of $GL(2)$. One can compute functions of the K-jet which are invariant to different subgroups, so called differential invariants of $I(x, y)$. For example, consider

$$\mathcal{F}^r(x, y) = \begin{bmatrix} I \\ I_x I_x + I_y I_y \\ I_{xx} I_x I_x + 2 I_{xy} I_x I_y + I_{yy} I_y I_y \\ I_{xx} + I_{yy} \\ I_{xx} I_{xx} + 2 I_{xy} I_{xy} + I_{yy} I_{yy} \\ I_{xxx} I_y I_y I_y - 3 I_{xxy} I_x I_y I_y + 3 I_{xyy} I_x I_x I_y \\ - I_{yyy} I_x I_x I_x \\ I_{xxx} I_x I_y I_y - 2 I_{xxy} I_x I_x I_y + I_{xxy} I_y I_y I_y \\ - 2 I_{xyy} I_x I_y I_y + I_{xyy} I_x I_x I_x + I_{yyy} I_x I_x I_y \\ I_{xxx} I_x I_x I_y + 2 I_{xxy} I_x I_y I_y - I_{xxy} I_x I_x I_x \\ - 2 I_{xyy} I_x I_x I_y + I_{xyy} I_y I_y I_y - I_{yyy} I_x I_y I_y \\ I_{xxx} I_x I_x I_x + 3 I_{xxy} I_x I_x I_y + 3 I_{xyy} I_x I_y I_y \\ + I_{yyy} I_y I_y I_y \end{bmatrix}$$

To compare two feature points \mathbf{p}^1 and \mathbf{p}^2 detected in two images, which are described by a feature vectors \mathbf{f}^1 and \mathbf{f}^2 (feature vector \mathbf{f}^i could be computed from $\mathcal{F}(\mathbf{p}^i)$ or $\mathcal{F}^r(\mathbf{p}^i)$), we determine their Mahalanobis distance

$$d(\mathbf{p}^1, \mathbf{p}^2) = (\mathbf{f}^2 - \mathbf{f}^1)^t \Sigma^{-1} (\mathbf{f}^2 - \mathbf{f}^1). \quad (1)$$

For a set of landmarks $\mathcal{P} = \{\mathbf{p}_j\}$, this suggests a distinctiveness measure for a landmark $\mathbf{p}_i \in \mathcal{P}$.

$$\delta(\mathbf{p}_i) = \min_{\mathbf{p}_j \in \mathcal{P}, \mathbf{p}_i \neq \mathbf{p}_j} d(\mathbf{p}_i, \mathbf{p}_j) \quad (2)$$

The set of candidate landmarks can be sorted by $\delta(\mathbf{p}_i)$ where the most distinctive landmark has the largest value of $\delta(\mathbf{p}_i)$.

For example, in previous work [3, 9] we developed a pre-selection method which used color cues to select markers. The color regions sought by the algorithm could be characterized as a *picture in a frame*, meaning sharply delimited. The features selected by the algorithm tended to be uniquely colored objects surrounded by featureless areas or general clutter.

In next fig. 1, we present a landmark selection and its robustness, some mismatches are easily recognized using epipolarity, even if some mistakes appears, at least all the other landmarks has been recognized. This will be important for visual servoing purposes.

2.2 Visual servoing

Visual servoing[1, 8] is the fusion of results from many elemental domains including image processing (high-speed), kinematics, dynamics, control theory, and real time computing. The task in visual servoing is to control a robot to manipulate its environment using vision as opposed to just observing the environment.

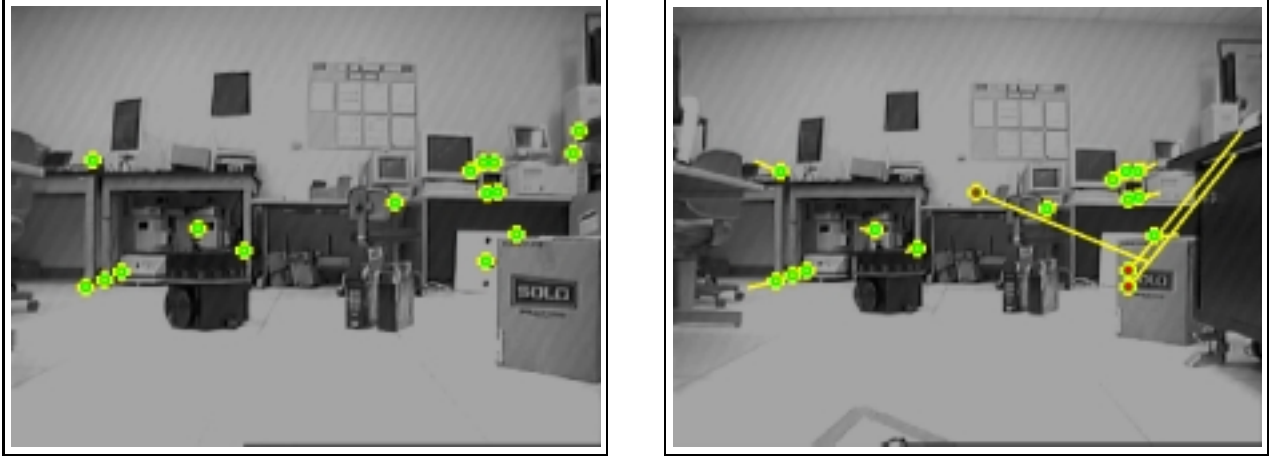


Fig. 1. Two images in a sequence: Landmarks were selected in the upper image and recognized in the lower image. The lines denote correspondences. Note that in this pair, three mismatches occurred.

Embedding visual servoing in the task function approach allow us to take advantage of general results helpful for the analysis and the synthesis of efficient closed loop control schemes. Methods for non-holonomic robots using visual servoing have been developed by many authors[10, 19].

Suppose that an observed landmark has image coordinates $l_i=(u, v)^t$ and external coordinates $P_i=(X, Y, Z)^t$ expressed in the camera frame. The point P_i and its projection are related by:

$$\begin{pmatrix} u \\ v \end{pmatrix} = \frac{1}{Z} \begin{pmatrix} X \\ Y \end{pmatrix} \quad (3)$$

It follows that the velocity of the projection $\dot{\mathbf{l}}_i$ due to the robot motion $\mathbf{v}=\dot{\mathbf{r}}$ is:

$$\dot{\mathbf{l}}_i = \begin{pmatrix} -\frac{1}{Z} \frac{u}{Z} & -\frac{1}{Z} \frac{v}{Z} & -(1+u^2) \\ 0 & \frac{v}{Z} & -uv \end{pmatrix} \dot{\mathbf{r}} = J_i(u, v, z)\mathbf{v} \quad (4)$$

This is a planar version of the so-called Image Jacobian expressed as a function of observed values u and v and the unknown Z . More generally, if \mathbf{S}_r (who has $l_{i_1}, l_{i_2}, \dots, l_{i_n}$) is comprised of landmarks with image coordinates $\{l_i\}$, the evolution of \mathbf{S}_r as a function of the motion of the system can be written by:

$$\dot{\mathbf{S}}_r = \mathbf{J}\mathbf{v} \quad (5)$$

where \mathbf{J} depends now on the image coordinates and depth of every observed point. Since the motion of the system is already stabilized by encoder feedback, it is usually possible to model system dynamics as a pure time delay and to choose a control input $\mathbf{u} = (\dot{x}, \dot{y}, \dot{\theta})$. Under these conditions, given a fixed set point $\mathbf{S}^* = \mathbf{S}_r(s)$, feedback system of the general form will (without noise) uncertainty about \mathbf{J} and the given dynamics, be locally asymptotically stable for an appropriate choice of the gain k :

$$\mathbf{u}(t) = -k(\mathbf{J}^t\mathbf{J})^{-1}\mathbf{J}^t(\mathbf{S}_c(t) - \mathbf{S}^*) \quad (6)$$

There are some ways to implement this controller, in our case the fact that the results of eq. 6 deteriorate rapidly when the orientation difference between the controlled system and the reference are large. However, recall that we can easily compute θ_r (robot orientation at r) using epipolar methods and we can use this value to rotate observed data into the frame of the reference trajectory. In practice, we apply to the modified values, and adjust the control policy for ω to be:

$$\omega(t) = K(\omega_r(t) + \theta_r(t)) - \eta K \dot{x} \quad (7)$$

we then *tune* the controller for a nominal capture region.

3 Experimental results

The method described above has been implemented at the University of Illinois at Urbana-Champaign and Johns Hopkins University. Part of this results has been tested at Fort Sam Huston (San Antonio, Texas, USA).

3.1 Our system

Inside the robot we have a linux system and a PVM structure that allows to work several process (or servers) at same time, basically feature tracking, feature prediction, feature selection, image-based learning, pan-tilt head controllers and robot controllers, and some of this servers works using our XVision platform. Image processing is performed at video rate using XVision.

In order to perform a visual-based navigation task, we use the following architecture system (Fig. 2) inside our non-holonomic robot (Fig. 3):

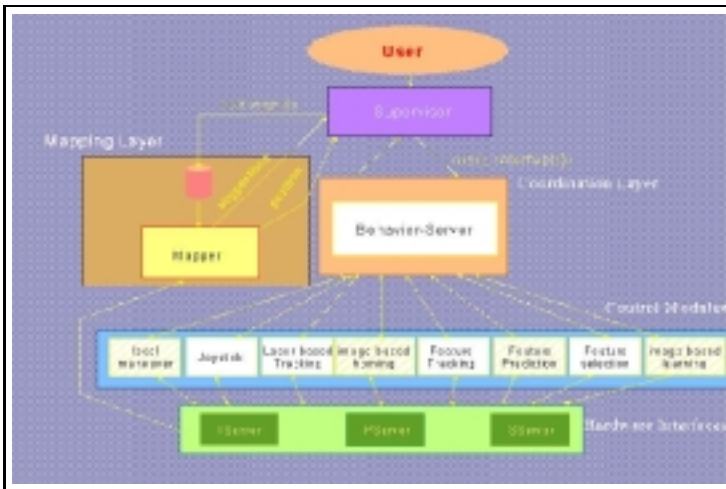


Fig. 2. Architecture



Fig. 3. Galaxian

Each box in the bottom of fig. 2 represent a server (PServer, TServer, SServer), and these are connected to the relative modules (like: tracking, mapping, selecting landmarks, etc). Our robot is equipped with a color camera, a pan-tilt head, a SICK laser scanner, GPS; in this demo, laser and GPS was not used.

3.2 Problem

The basic idea it will be to perform navigation tasks using a map representation. These tasks will be specified in terms of the information that is contained in the map.

Map-based navigation requires the ability to execute motions relative to visible landmarks, and also the ability to switch the focus of attention as landmarks come into and pass out of view. For example, navigating a corridor might involve moving down a hallway to a corner by executing visual servo controlled motions using some visible landmark to specify the navigation goal; at a corner, a new landmark might become visible, and the controller would then switch to this new landmark for the next visual servo controlled motion. Thus, we envision a high-level controller that invokes lower level visual servo controllers to execute individual segments of the trajectory.

First of all, once that natural landmarks has been selected (see fig. 4), using the method described in §

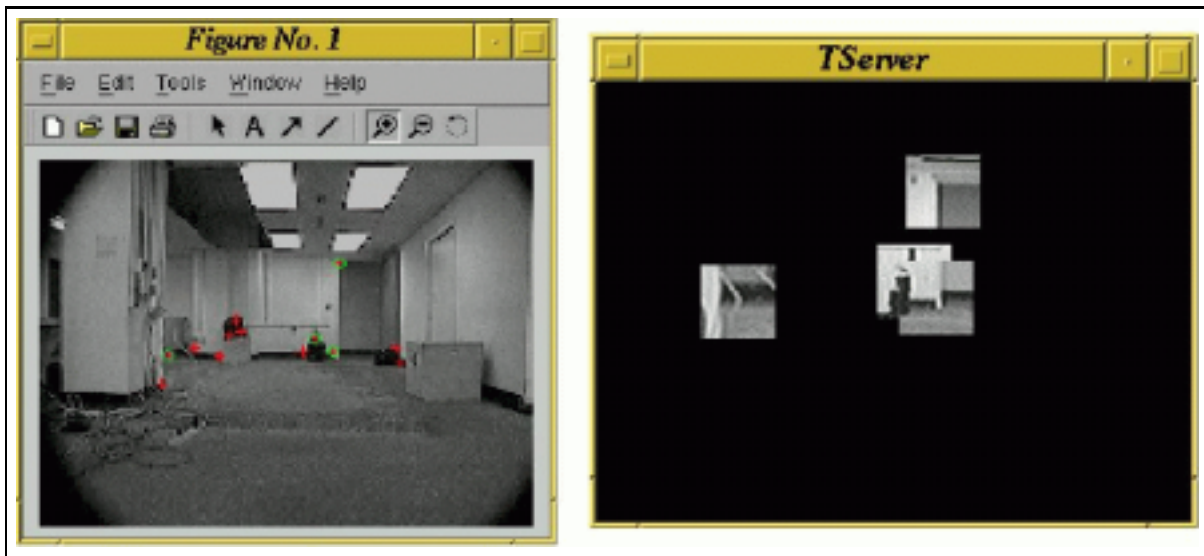


Fig. 4. Initialization step: landmarks acquisition



Fig. 5. Generating a path

2.1, our system generate a path (manually) where all the visual trajectories of tracked landmarks are stored (see fig. 5).

Then, robot uses visual servoing techniques (described in § 2.2) to follow original trajectory backwards or forwards to repeat the same trajectory as many times as will be possible (see fig. 6, 7)². In this case, four landmarks has been automatically selected, then a path as been executed and our robot achieve and repeat the path even if the ground conditions was not the best.

Some relevant information is that landmarks has been robust during this trajectory (this prove the effectiveness of our method), and in many parts, it wasn't a ground plane (carpet ground) and in some places carpet was very damaged, so, in order to repeat several times the path, our visual servoing procedure

² Visit this place <http://www-cvr.ai.uiuc.edu/TMR/> to see more information and some MPEGs



Fig. 6. Generating a path: First Time



Fig. 7. Generating a path: Reply movement

proves its robustness.

3.3 Updating landmarks

During subsequent navigation, the image motion of observed feature trajectories provides direct feedback for robot motion control. As features leave the robot field of view, new features are acquired, these features are tracked and used to control robot motion.

From a single location, we can accurately obtain the relative location of these landmarks. When the robot moves to a nearby location and detects landmarks, some will be new and some will have been seen previously. Those landmarks seen from both locations serve as a link to be able to infer the relative position of the other landmarks in this pair. The corresponding graph whose nodes are landmarks and whose arcs correspond to landmarks seen from one location will naturally be sparse since there is only a small number of landmarks observed in each image, and the representation is expected to grow linearly in the number of landmarks. When landmark pairs are subsequently observed from additional viewpoints, the estimates of their spatial relationship can be updated – depending upon the uncertainty representation, there are a variety of probabilistic or set-based techniques [3]. Most steps in a navigation task only require using those landmarks that are in close proximity to the robot, and navigation planning generally requires searching through steps, each of which only requires inference using a local subset of the landmarks. While this subset may be densely connected, it will generally be small enough that computations and inference (e.g. uncertainty propagation) are reasonable.

Landmarks will move in and out of view. Thus, it is also necessary to anticipate when a previously observed marker may come into view. Updating new landmarks in our system has been done using an affine

transformation method[2]. It was necessary to use this method in order to follow landmarks. It is necessary to acquire a new image when one of the tracked landmarks is on the way to disappear (crossing the threshold established into the image), but as our robot is in movement during the acquisition, the position of the new landmark is not exactly the same. In our system, we usually track 4 landmarks, suppose that after landmark X disappear we acquire a new image in time t and this landmark will be P but now in a the time $t+1$, as we know the others landmarks position during this process O, A, B , is quite easy to obtain the new position P' using an affine transformation method.

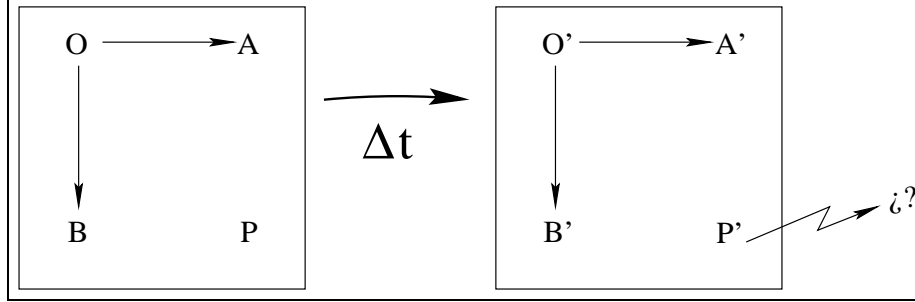


Fig. 8. Landmark configuration: Updating

In order know the actual position of new landmark into the image an affine transformation implementation has been done using following equation:

$$\begin{aligned}\bar{P} &= (A - O)\alpha + (B - O)\beta \\ \bar{P}' &= (A' - O')\alpha + (B' - O')\beta + O'\end{aligned}\quad (8)$$

where \bar{P} and \bar{P}' are the distances between O and the respective P . α and β are computed by:

$$\begin{aligned}\alpha &= \frac{(P_x - O_x)(B_y - O_y) - (P_y - O_y)(B_x - O_x)}{(A_x - O_x)(B_y - O_y) - (B_x - O_x)(A_y - O_y)} \\ \beta &= \frac{(P_y - O_y)(A_x - O_x) - (P_x - O_x)(A_y - O_y)}{(A_x - O_x)(B_y - O_y) - (B_x - O_x)(A_y - O_y)}\end{aligned}\quad (9)$$

Final landmark computation of P' is given by:

$$\begin{aligned}P'_x &= \bar{P}'_x + O'_x \\ P'_y &= \bar{P}'_y + O'_y\end{aligned}\quad (10)$$

Is important to remember that, new landmark acquisition should be as fast as we can in order to ensure a good computation time of all landmarks. This updating procedure is done during the entire robot movement. All the information generate during this motion is stored on the way to reproduce the path many times. We use again the affine transformation method in order to obtain the inverse procedure (to find where was the final landmark position before disappear), this information is relevant to update and find again an old landmark.

4 Summary & conclusions

We have presented a method for selecting and recognizing salient landmarks based on the method of Schmid and Mohr [15] and a criterion for selecting the most distinctive landmarks. The selection of the most recognizable landmarks can be important in real-time applications where it is not feasible to track all possible landmarks. Experimental results have characterized the performance of the method on indoor scenes where a mobile robot might typically operate. Then, these landmarks has been used in order to achieve a visual-based task with success. The problem of working with a variable set of landmarks is interesting and relevant, and the fact that the sensors has a limited working range justify this research topic.

Unfortunately, distinctive landmarks are more readily recognized, but in some case the most distinctive landmark correspond to viewpoint-dependent features such as T-junction/occlusion boundaries.

Future work will be on the way to use this landmarks to construct a representation of the environment (i.e., the mapping problem); annotating the map with landmarks that can be recognized in later phases of operation, and that can be used for robot localization; and finally, using the newly constructed map for navigation. Navigation tasks will be defined by high-level goals that are specified in terms of the landmarks that are stored in the map. For example, we might specify that the robot should *move down the corridor along the wall and enter the first door on the right*. To move to a location where a specific set of landmarks is in view, the map can be searched for a path between the current set of visible landmarks to the goal.

Acknowledgments

This research was supported by the National Science Foundation under IRI-9711967 and by DARPA under DAAE07-98-C-L031.

References

1. B. Espiau, F. Chaumette, and P. Rives. A new approach to visual servoing in robotics. *IEEE Transactions on Robotics & Automation*, Vol. 8(Num. 3), June 1992.
2. O. Faugeras. Three-Dimensional Computer Vision: A Geometric Viewpoint. *The MIT press*, Cambridge, Massachusetts, USA, 1993.
3. G.D. Hager, E. Yeh D.J. Kriegman, and C. Rasmussen. Image-based prediction of landmark features for mobile robot navigation. In *Proceedings of the IEEE International Conference on Robotics & Automation (ICRA)*, pages 1040–1046, New Mexico, USA, April 1997.
4. G. Hager, D. Kriegman, A. Georghiades, and O. Ben-Shahar. Toward domain-independent navigation: Dynamic vision and control. In *IEEE Conf. on Decision & Control*, 1998.
5. G. Hager and K. Toyama. XVision: A portable substrate for real-time vision applications. *Computer Vision & Image Understanding*, 69(1):23–37, 1998.
6. C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, pages 147–151, 1988.
7. R. Hartley. Lines and points in three views and the trifocal tensor. *Int. J. Computer Vision*, 22(2):125–140, 1997.
8. S. Hutchinson, G. Hager, and P. Corke. A tutorial on visual servo control. *IEEE Trans. on Robotics & Automation*, 12(5):651–670, 1996.
9. M. Knapek, R. Swain-Oropeza and D.J. Kriegman. Selecting Promising Landmarks. In *the Int. Conf. on Robotics & Automation (ICRA)*, San Francisco, USA, April 2000.
10. J. Kosecka and R. Bajcsy and M. Mintz. Control of Visually Guided Behaviors. In *Real-time Computer Vision*, Cambridge Press, Ed. Christopher Brown and Demetri Terzopoulos, 1994.
11. D. Kriegman, G. Hager, and A. Morse. *The Confluence of Vision & Control*. Springer-Verlag, 1998.
12. S. Li and S. Tsuji. Finding landmarks autonomously along a route. In *Int. Conf. on Pattern Recognition*, pages 316–319, 1992.
13. H. P. Moravec. The Stanford cart and the CMU rover. *Proc. of the IEEE*, 71(7), July 1983.
14. A. J. Munoz and J. Gonzalez. Two-dimensional landmark-based position estimation from a single image. In *Proc. IEEE Int. Conf. on Robotics & Automation (ICRA)*, 1998.
15. C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. *IEEE Trans. Pattern Anal. Mach. Intelligence*, 19(5):1997, May 1997.
16. C. Schmid, R. Mohr, and C. Bauckhage. Comparing and evaluating interest points. In *Int. Conf. on Computer Vision (ICCV)*, 1998.
17. S. Simhon and G. Dudek. Selecting targets for local reference frames. In *Proc. IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 2840–2845, 1998.
18. D. Slater and G. Healey. The illumination-invariant recognition of 3d objects using local color invariants. *IEEE Trans. on Pattern Analysis & Machine Intelligence*, 18(2):206–210, Feb. 1996.
19. R. Swain-Oropeza, M. Devy and V. Cadenat. Controlling the Execution of a Visual Servoing Task. In *Journal of Intelligent and Robotic Systems : Theory and Applications (Incorporating Mechatronic Systems Engineering)*, Vol. 25 (Num. 4), June 1999.
20. Y. Takeuchi, P. Gros, M. Hebert, and K. Ikeuchi. Visual learning for landmark recognition. In *Proc. Image Understanding Workshop*, pages 1467–1473, 1997.

21. S. Thrun. Finding landmarks for mobile robot navigation. In *IEEE Conf. on Robotics & Automation (ICRA)*, pages 958–963, 1998.
22. C. Tomasi and J. Shi. Good features to track. In *Proc. IEEE Conf. on Comp. Vision & Patt. Recog. (CVPR)*, pages 593–600, 1994.
23. E. Yeh and D. Kriegman. Toward selecting and recognizing natural landmarks. In *IEEE Int. Workshop on Intelligent Robots & Systems*, pages 47–53, 1995.
24. Z. Zhang, R. Deriche, O. Faugeras, and Q.-T. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. Technical report, INRIA, 1994.