Omni-directional Vision for Robot Navigation

Niall Winters[1], José Gaspar[2], Gerard Lacey[1], José Santos-Victor[2]

[1]Computer Vision and Robotics Group,
Department of Computer Science,
University of Dublin, Trinity College,
Dublin 2 - Ireland.
{Niall.Winters, Gerard.Lacey}@cs.tcd.ie

[2]Instituto de Sistemas e Robótica,
Instituto Superior Técnico,
Av. Rovisco Pais, 1,
1049-001 Lisboa - Portugal.
{jag, jasv}@isr.ist.utl.pt

Abstract

We describe a method for visual based robot navigation with a single omni-directional (catadioptric) camera. We show how omni-directional images can be used to generate the representations needed for two main navigation modalities: Topological Navigation and Visual Path Following.

Topological Navigation relies on the robot's qualitative global position, estimated from a set of omni-directional images obtained during a training stage (compressed using PCA). To deal with illumination changes, an eigenspace approximation to the Hausdorff measure is exploited. We present a method to transform omni-directional images to Bird's Eye Views that correspond to scaled orthographic views of the ground plane. These images are used to locally control the orientation of the robot, through visual servoing.

Visual Path Following is used to accurately control the robot along a prescribed trajectory, by using bird's eye views to track landmarks on the ground plane. Due to the simplified geometry of these images, the robot's pose can be estimated easily and used for accurate trajectory following.

Omni-directional images facilitate landmark based navigation, since landmarks remain visible in all images, as opposed to a small field-of-view standard camera. Also, omni-directional images provide the means of having adequate representations to support both accurate or qualitative navigation. Results are described in the paper.

1. Introduction

In recent years visually-guided navigation [19] has been approached in many different ways. Map-building using image sequences from standard single or stereo cameras has been explored in [2, 20]. Kosaka and Kak [18] combined model based reasoning and Kalman filtering, which requires a 3D geometric model of the environment. In [17], sonar data is collected along corridors and used together with visual data (simply vertical edges) to aid navigation. Although previous research produced many interesting results, most systems required a large amount of computational power and still lacked the robustness required for many real-world applications.

More recently, omni-directional cameras have been used for navigation. This idea has much in common with biology where the majority of insects and arthropods benefit from a wide field of view [26]. Aihara et al. [1] use a sequence of autocorrelated omni-directional images to determine global position. However, as there is no simultaneous control of local pose, they are forced to densely sample the environment with many images. Mobile robot self-localisation was studied by [4], taking data from a colour omni-directional camera as the input to a neural network, so the robot could learn its environment. This system relied upon differently coloured artificial landmarks in a *test* scene. Delahoche [8] presented a system which merged odometry and omni-directional image data using an Extended Kalman Filter. The position of beacons in the absolute reference frame of the environment were known *a priori*.

Appearance based methods for navigation [13, 15, 21] too, have been an active area of research. In [25], appearance based methods and visual servoing are combined but the image geometry, matching scheme and servoing strategy were different from those detailed in this paper.

Our work focuses on the differing nature of the navigational requirements (and environmental representations) when covering long distances, as compared to those for short paths. Again, if we look to biology, many animals make alternate use of landmark-based navigation and (approximate) route integration methods [28]. For example, we can walk along a city avenue, without ever knowing our precise position along the way. However, to enter our hall door, our movements demand more precision.

This *path distance/accuracy tradeoff* between long-distance/low-precision and short-distance/high-accuracy mission segments plays an important role in finding efficient solutions to the robot navigation problem. In this paper, we denote these navigational modes as *Topological Navigation* versus *Visual Path Following*.

Topological Navigation combines appearance based methods [16, 27] to estimate the robot's qualitative *global* position, and visual servoing to control *locally* the robot heading [29]. *Visual Path Following* is used for precise control of the robot along a trajectory pre-specified in image coordinates, based on the landmarks tracked in the omni-directional images [10].

Omni-directional images provide simple and adequate representations to support these different navigation modes. Additionally, the use of omni-directional images implicitly allows us to deal with partial occlusion and by using a view-based approximation to the Hausdorff measure we can cope with illumination changes. All these aspects are illustrated by the complete experiments described in this paper.

Section 2 describes our omni-directional vision system and the method to unwarp omni-directional images to bird's eye (orthographic) views of the ground plane, which greatly simplifies the navigation problem. Section 3 is devoted to the problem of navigating using eigenspaces as topological maps. In Section 4, we introduce the Visual Path Following approach to control the robot motion, while in Section 5 we present results from experiments combining both navigation strategies. Finally, in Section 6 we draw some conclusions and establish future directions of research.

2 The Omni-directional Vision System

The only sensor we use for navigation is an omni-directional catadioptric camera [3], composed of a CCD camera upward looking at a *spherical* mirror[1]. The camera is mounted on top of the mobile platform with its axis placed coincident to the platform's rotation axis (see Fig.1).

(a) (b)

Figure 1. (a) Omni-directional camera. (b) Camera mounted on a Labmate mobile robot.

In general, 3D lines in the environment are projected as curves in an omni-directional image. However, we can use models to correct some of the distortions and obtain the two main types of unwarped images: panoramic images [5] and *Bird's Eye Views*.

The *Bird's Eye View* is an orthographic projection of the ground plane (e.g., corridors appear as image bands of constant width). This kind of image may be obtained by radial correction around the image centre[2].

2.1 Bird's Eye View of the Ground Plane

Points in the 3D space, P, are projected as image points, p, by means of a projection operator, $\mathcal{P}$:

$$p = \mathcal{P}(P, \theta) \tag{1}$$

where θ contains all the intrinsic and extrinsic parameters of the catadioptric panoramic camera:

$$\theta = [L\ f\ u_o\ v_o]^T .$$

The mirror radius can be measured easily, but the camera-mirror distance, L, focal length, f and principal point, (u_0, v_0), can only be determined up to some error:

$$\delta\theta = [\delta L\ \delta f\ \delta u_0\ \delta v_0]^T .$$

To estimate $\delta\theta$ we use a set of known 3D points, P^i, and the corresponding image projections p^i, then minimize the following cost function:

$$\delta\theta = \arg\min_{\delta\theta} \sum_i \parallel p^i - \mathcal{P}(P^i, \theta_0 + \delta\theta) \parallel^2 \tag{2}$$

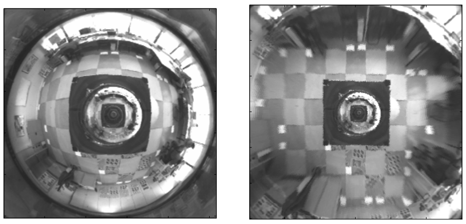


Figure 2. Left: Original omni-directional image. Right: Remapped bird's eye view image.

This procedure defines a mapping between radial distances measured in the ground plane and the respective image co-ordinates, and can be efficiently implemented by means of a look-up table. It allows us to unwarp omni-directional images to *bird's eye views* - see Figure 2. For a more detailed description of the omni-directional vision system and the images unwarping techniques, see [10].

[1] Although our sensor does not have a single projection centre as in [3] we found that this is not a severe limitation to our approach.

[2] This unwarping is done by software. Hicks [12] has shown how to obtain ground plane unwarped images, directly from a custom-shaped mirror.

3 Navigating using Topological Maps

When walking in a city, humans do not rely on metric knowledge, which is a good analogy to topological navigation. We adopt a similar approach for robot navigation, to accomplish missions such as *go to the third office on the left-hand side of the second corridor*. The behaviours required in this case must allow the robot to travel along a corridor, recognise the ends of a corridor, make turns and identify (and count) door frames. These behaviours are implemented by means of an appearance based system and a visual servoing strategy.

The topological map is a graph that allows the determination of the robot's global *qualitative* position within its environment. *Nodes* represent places (identified by images) where actions are undertaken (e.g. turn left). To navigate along *links*, we assume that some environmental image features can be used for servoing (e.g. follow the road).

3.1 Position Estimation

The number of images required to represent the environment is very large and one needs to compress this information. We used a *low dimensional eigenspace* representation [22], constructed using Principal Component Analysis. This image eigenspace and their topological relations form the map we use for navigation.

The problem of estimating the robot position in the context of our topological map is that of determining the reference image, $\mathbf{I_k}$ that best matches the current image, $\mathcal{I}$ within the low dimensional eigenspace. Each image, $\mathbf{I_k}$, in the initial training set is associated with a *qualitative* robot position in the topological map (e.g. half way along the corridor).

Each acquired image, $\mathcal{I}$, is first projected into the eigenspace to obtain the corresponding coefficient vector, $\mathcal{C}$. A *real time* comparison between this image and all the reference images of the topological map is computed as follows:

$$d_k = \frac{\sum_{j=1}^{M} {\rho_j^k}^2 \lambda_j}{\sum_{j=1}^{M} \lambda_j} \qquad \text{where} \qquad \rho_j^k = \left| c_j - c_j^k \right|$$

The value of d_k expresses the distance between the current image, $\mathcal{I}$ and the various reference images, $\mathbf{I_k}$. The value of ρ_j^k expresses the distance between images $\mathcal{I}$ and $\mathbf{I}_k$ along the j^{th} direction of the eigenspace. This distance is weighed by the eigenvalues λ_j to express the relative importance of this direction in the entire eigenspace. The position of the robot is that associated with the basis image, $\mathbf{I}_k$, which yields the lowest distance, d_k.

3.2 Dealing with Illumination Changes

When using intensity images to build a topological representation of the environment the robot is prone to miscalculating its location where *large non-uniform* deviations in illumination occur (see Fig.3).

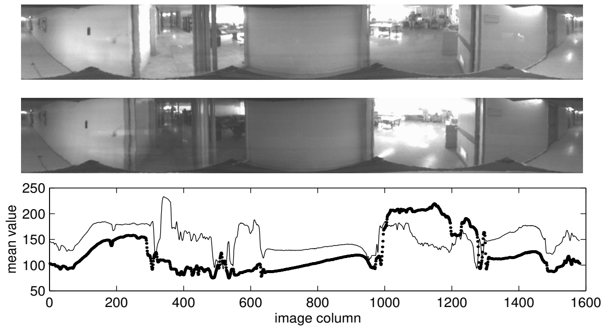


Figure 3. Top: images acquired at 5pm and 11am. Bottom: image intensity shows large non-uniform deviation in brightness (the thin line relates to the first image).

This can be overcome by using edge images to represent the environment. Matching is achieved by using an eigenspace approximation to the *Hausdorff Fraction* [14]. The Hausdorff distance [24] (of which the Hausdorff fraction is a subset) is a technique whereby one can measure the distance between two sets of points, in our case edge images. The eigenspace approximation is built as follows: Let I_m be an observed edge image and I_n^d be an edge image from the topological map, arranged as column vectors. The Hausdorff fraction, $\hbar(I_m, I_n^d)$, which measures the similarity between these images, can be written as:

$$\hbar(I_m, I_n^d) = \frac{I_m^T I_n^d}{||I_m||^2}$$

An image, I_k can be represented in a low dimensional eigenspace [22, 29] by a coefficient vector, $\mathcal{C}_k = [c_1^k, \cdots, c_M^k]^T$, as follows:

$$c_j^k = e_j^T.(I_k - \bar{I}).$$

Here, $\bar{I}$ represents the average of all the intensity images and can be also used with edge images. Thus, the eigenspace approximation to the Hausdorff fraction can be efficiently computed as:

$$\widehat{\hbar}(I_m, I_n^d) = \frac{C_m^T C_n^d + I_m^T \bar{I} + I_n^{dT} \bar{I} - ||\bar{I}||^2}{||I_m||^2}$$

To test this view-based approximation we collected a sequence of images, acquired at different times, 11am and

5pm, near a large window. Figure 4 shows the significant changes in illumination, especially near the large window at the bottom left hand side of each omni-directional image. Even so, the view based approximation can correctly determine that the unknown image shown in Figure 4(a) was closest to the database image shown in Figure 4(b), while PCA based on brightness distributions would fail. For completeness, Figure 4 (c) and (d) shows a run-time edge image and its corresponding retrieved image using the eigenspace approximation to the Hausdorff fraction.

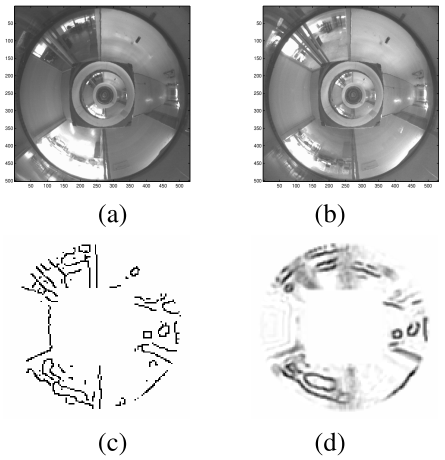


Figure 4. (a) An omni-directional image obtained at 11:00, (b) one obtained at 17:00; (c) An edge-detected image and (d) its retrieved image.

Figure 5 shows the robot's qualitative position estimated over time, using both gray-level distributions (left) and the Hausdorff fraction (right). The images for the topological localisation were acquired at 11am and experiments were conducted at 12pm and 5pm. Only when using the Hausdorff fraction did we obtain similar results, independently of the illumination conditions.

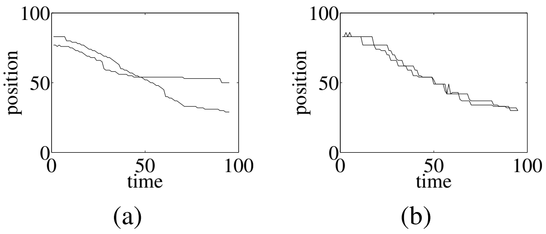


Figure 5. Position estimation with large non-uniform illumination changes (a) using brightness distributions and (b) the Hausdorff fraction.

3.3 Dealing with Occlusions

A major advantage of using omni-directional images is that we can deal with a relatively *dynamic* environment, where people partially occlude the robot's view. When a person is as close as possible to the robot (see Figure 6), the occlusion is not sufficiently large so as to cause the robot to misinterpret its topological position. Using the low dimensional eigenspace approach the closest image is still correctly determined. An alternative approach is taken by [6], where a number of small image windows (e.g., 9 x 9 pixels) are used to cope with occlusion. The success of their method depended on the number and size of windows used.

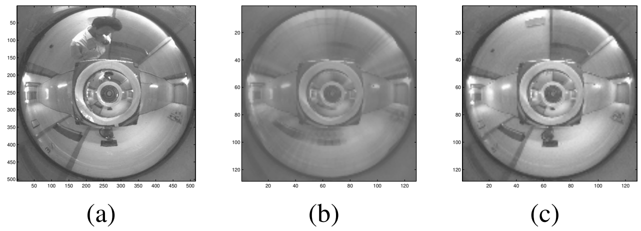


Figure 6. (a) A person close to the robot. The retrieved image (b) and the input image corresponding to the estimated position in the topological space (c).

3.4 Corridor Following Behaviour

The remaining problem to be solved for effective navigation using the topological map is that of defining a suitable vision-based behaviour for corridor following. Since most corridors have parallel guidelines, we can exploit this information to drive the robot along the corridor.

Visual feedback from the omni-directional camera is used to keep the robot centered in the corridor. This visual feedback is in the form of the extracted corridor guidelines, from *bird's eye view* images. The servoing task is significantly simplified, because these images become a scaled orthographic projection of the ground plane co-ordinates. We use a simple kinematic planner to control *locally* the robot position and orientation in the corridor, while maintaining a forward trajectory. Details on the control law can be found in [29].

In order to track the corridor guidelines, we first find edges within pre-defined bounding boxes and then perform robust line fitting using a procedure based on RANSAC [9]. Predicting the position of these bounding boxes is a determinant for the robustness of the tracker, which could otherwise start tracking door frames and other image line segments.

Figure 7 shows a sequence of bird's eye view images acquired during tracking. The prediction is very accurate and vastly improves the probability of extracting the corridor guidelines rather than erroneous data such as door frames.

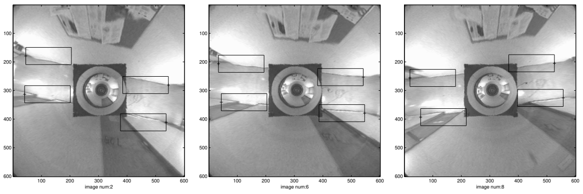


Figure 7. Ground plane view of the robot's orientation and translation over time.

4 Visual Path Following

In this section, we outline the role of omni-directional vision for *Visual Path Following*, used for local, precise navigation. Examples of such situations include door traversal, docking and navigating inside rooms or very cluttered environments, where a reference trajectory must be followed accurately.

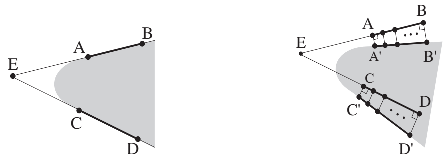
4.1 Feature tracking and self-localisation

We utilise bird's-eye view images to track environmental features so as to estimate the robot's position/orientation or to drive the robot along a given trajectory.

The features selected for tracking were image corners defined by the intersection of edge segments [11], which can usually be found in indoor environments. The detection process benefits from a larger spatial support, as opposed to local corner detection filters, thus leading to increased accuracy and stability. Figure 8(a) shows a corner point E defined as the intersection of lines $\overline{AB}$ and $\overline{CD}$. In this way, corners do not necessarily have to correspond to image points of extreme changes in brightness. This approach can deal with information loss due to occlusion or filtering (e.g. the "roundness" of corners due to image smoothing).

We track corner points by tracking the corresponding support edge lines. Each edge segment is tracked by searching along its perpendicular direction and using local photometric and geometric criteria, combined with a robust fitting procedure [9]. Figure 8(b) demonstrates how the segments $\overline{AB}$ and $\overline{CD}$ at time t, are tracked to the segments $\overline{A'B'}$ and $\overline{C'D'}$, respectively, in the image at time $t + 1$.

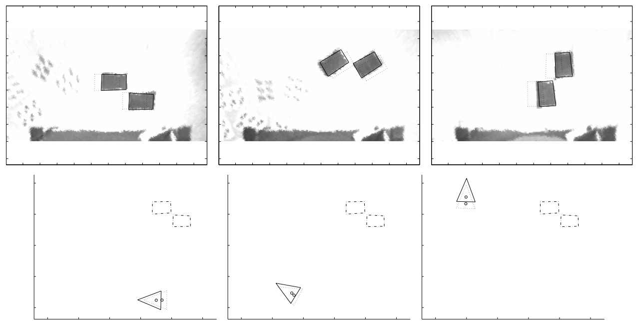
After determining the new corner positions, we estimate a 2D rigid transformation between two successive bird's eye views, yielding the robot position and orientation relative to some pre-defined co-ordinate system.



(a) (b)

Figure 8. (a) Corner point E is defined as the intersection of lines $\overline{AB}$ and $\overline{CD}$; (b) Segments $\overline{AB}$ and $\overline{CD}$ are adjusted to $\overline{A'B'}$ and $\overline{C'D'}$.

Figure 9 illustrates the tracking (top) and pose estimation (bottom) process, using a known pattern located on the ground plane as a navigational landmark. The chosen landmark is defined by eight line segments which intersect at eight corner feature points. At the current stage of implementation, the relevant features to track and the feature coordinate system are initialised by the user. In order to detect the loss of tracking during operation, the tracking process is continuously self-evaluated by the robot. This evaluation is based on gradient intensities obtained within specified areas around the landmark edges. If these gradients decrease significantly compared to those expected, a recovery mechanism is immediately launched.



Displayed area: 300pix by 230pix.

Figure 9. Top: feature tracking at three time instances (dashed and solid lines show the landmark's original and final positions). Bottom: robot pose calculated over time, expressed in the world co-ordinate system defined by the landmark.

Given the robot pose and a reference trajectory, we designed a control scheme that drives the distance and orientation errors to zero, while maintaining a forward velocity (see [7, 10] for details).

5 Experimental Results

All experiments were undertaken with a TRC Labmate from HelpMate Robotics Inc., equipped with an omni-directional vision system (Figure 1(b)) built in-house. Greyscale images were captured with a full resolution of 768x576 pixels, and sub-sampled to 128x128 images for PCA and 600x600 for visual servoing and visual path following. All the processing was carried out on-board the Labmate mobile platform by a Pentium II 350MHz PC.

5.1 Navigating with a Topological Map

A series of experiments were conducted to test the navigational behaviours required to perform global missions using the topological map. The topological map was built with omni-directional images, acquired every 50 cm, along the corridors. At corresponding distances, bird's eye views were used for local pose control. Reference positions were ordered according to the direction of motion, thus maintaining a causality constraint.

The first tests show that appearance based methods can reliably provide qualitative estimates of the robot's position in the world, using a single corridor as an example. For this purpose we acquired a set of prior images, P, and ran the robot in the corridor to acquire a different set of run-time images, R. Figure 10 shows the distance d_k (see Section 3) between the prior and run-time images, P and R.

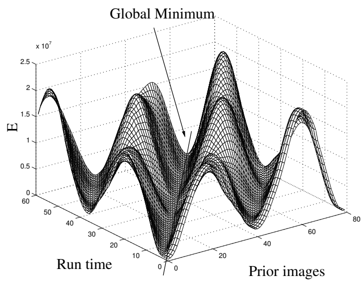


Figure 10. A 3D plot of images acquired at run-time, R versus those acquired a priori, P.

The *global* minimum, at each time instant, is the correct estimate of the robot's current topological position. The error surface degrades gracefully in the neighbourhood of the correct estimates. Local minima appear because some distant areas of the corridor look very similar, and can be avoided by restricting the search space to images close to the previously estimated position. To drive the robot along a central trajectory in the corridor, we used the servoing approach described in Section 3.

Figure 11 shows results from robot navigation along a section of the 7th floor at ISR. The distance traveled was approximately 21 metres. Odometry was used to display the path graphically.

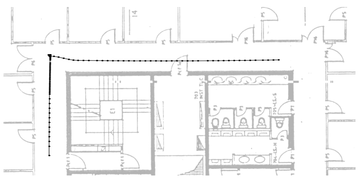


Figure 11. One of the paths traveled by the robot at our research institute. The total distance traveled was approximately 21 metres.

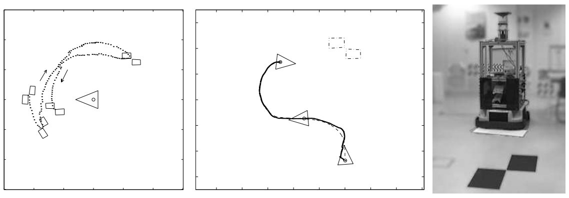
These results show that the proposed control scheme can successfully drive the robot along the corridor. When the robot arrives at the end of a corridor, it can switch to a different behaviour. In this example the behaviour launched at the end of each corridor is simply to perform a 90 degree turn, in order to proceed to the next corridor.

5.2 Visual Path Following Experiments

The *Visual Path Following* approach is used to navigate inside a room, which requires a more precise localisation and control of the robot trajectory, as compared to the previous example of long-distance missions.

A reference trajectory was specified in image coordinates, relative to a single landmark composed of two rectangles. The mobile robot moves under closed loop control, based on input from the omni-directional camera, as described in Section 4.

Figure 12(a) shows tracking results to illustrate the convenient use of omni-directional vision for landmark tracking, in spite of its large azimuthal movement relative to the robot. Figure 12(b) shows the reference trajectory (dashed line) and results of visual self-localisation (dotted line). Figure 12(c) shows the mobile robot after completion of the desired navigational task.



(a) (b) (c)

Figure 12. Visual path following experiment.

5.3 Integrated Experiments

The concluding experiment integrates global and local navigational tasks, by combining the *Topological Navigation* and *Visual Path Following* paradigms.

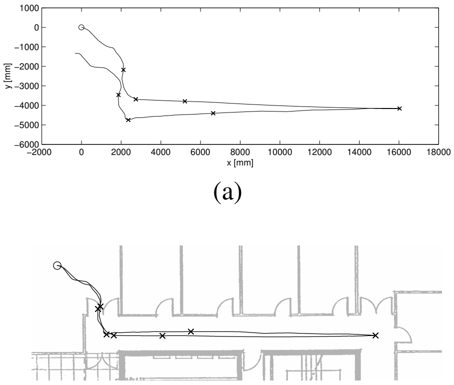
The mission starts in the Computer Vision Lab. Visual Path Following is used to navigate inside the Lab, traverse the Lab's door and drive the robot out into the corridor. Once in the corridor, control is transferred to the Topological Navigation module, which drives the robot all the way to the end of the corridor. At this position a new behaviour is launched, consisting of the robot executing a 180 degree turn, after which the topological navigation mode drives the robot back to the Lab entry point.

Figure 13. Experiment combining visual path following for door traversal and topological navigation for corridor following.

During this backward trajectory we use the same image eigenspaces as were utilised during the forward motion by simply rotating, in real-time, the acquired omni-directional images by 180 degrees. Alternatively, we could use the image's power spectrum or the Zero Phase Representation [23]. Finally, once the robot is approximately located at the lab entrance, control is passed to the Visual Path Following module. Immediately it locates the visual landmarks and drives the robot through the door. It follows a pre-specified path until the final goal position, well inside the lab, is reached. Figure 13 shows an image sequence to relate the robot's motion during this experiment.

In Figure 14(a) we used odometric readings from the best experiment to plot the robot trajectory. When returning to the laboratory, the uncertainty in odometry was approximately 0.5m. Thus, door traversal would not be possible without the use of visual control. Figure 14(b), shows the actual robot trajectory, after using ground truth measurements to correct the odometric estimates. The mission was successfully accomplished.

This integrated experiment shows that omni-directional images are advantageous for navigation and support different representations suitable both for Topological Maps, when navigating between distant environmental points, and Visual Path Following for accurate path traversal. Additionally, we have described how they can help in coping with occlusions, and with methods of achieving robustness against illumination changes.



(a)

(b)

Figure 14. A real world experiment combining Visual Path Following for door traversal and Topological Navigation for long-distance goals. Odometry results before (a) and after (b) the addition of ground truth measurements.

6 Conclusions & Future Work

In this paper, we presented a method for visual-based navigation using an omni-directional camera. One of the core observations is that different navigation problems have distinct requirements in terms of processing, accuracy and goals, that should be taken into account when designing an efficient navigation system.

We distinguished between missions that involved traveling long distances, without imposing strict requirements on the knowledge of robot position, and those where the robot must follow a pre-specified trajectory accurately: Topological Maps vs Visual Path Following.

Another key aspect is that omni-directional vision can offer the required representations to support these different navigation strategies. In particular, we described a method for obtaining a *bird's eye view* of the ground floor, that greatly simplified navigation problems, by removing perspective effects. Also, landmark based navigation is facilitated since landmarks remains visible in spite of significant rotations of the robot.

Integrated experiments combing both methodologies were presented. This integration is a powerful approach that, in our opinion, leads to an overall system which exhibits improved robustness, scalability and simplicity.

In the future, we will apply this methodology in more complex environments and address the problem of automatic extraction of relevant landmarks. Self-localisation will be improved to take full advantage of omni-directional images by using multiple widely separated landmarks.

Acknowledgements

This work has been partially funded by the EU TMR network SMART II and the project PRAXIS 2/2.1/TPAR/2074/95, SIVA.

References

[1] N. Aihara, H. Iwasa, N. Yokoya, and H. Takemura. Memory-based self-localization using omnidirectional images. In *ICPR'98*, pages 1799–1803, 1998.

[2] N. Ayache and O. Faugeras. Maintaining representations of the environment of a mobile robot. *IEEE Transactions on Robotics and Automation*, 5(6):804–819, December 1989.

[3] S. Baker and S. K. Nayar. A theory of catadioptric image formation. In *ICCV'97*, pages 35–42, January 1998.

[4] R. Cassinis, D. Grana, and A. Rizzi. Self-localization using an omni-directional image sensor. In *4th Int. Symp. on Intelligent Robotic Systems (SIRS'96)*, pages 215–222, Lisbon, Portugal, July 1996.

[5] J. S. Chahl and M. V. Srinivasan. Reflective surfaces for panoramic imaging. *Applied Optics (Optical Society of America)*, 36(31):8275–8285, November 1997.

[6] V. C. de Verdiere and J. L. Crowley. Local appearance space for recognition of navigation landmarks. In *6th Int. Symp. on Intelligent Robotic Systems (SIRS'98)*, pages 261–269, Edinburgh, United Kingdom, July 1998.

[7] C. C. de Wit, H. Khennouf, C. Samson, and O. J. Sordalen. Chap.5: Nonlinear control design for mobile robots. In Y. F. Zheng, editor, *Nonlinear control for mobile robots*. World Scientific series in Robotics and Intelligent Systems, 1993.

[8] L. Delahoche, C. Pégard, B. Marhic, and P. Vasseur. A navigation system based on an omnidirectional vision sensor. In *Proc. of the Int. Conf. on Intelligent Robotics and Systems 1997 (IROS'97)*, pages 718–724, Grenoble, France, 1997.

[9] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of ACM*, 24(6):381–395, June 1981.

[10] J. Gaspar and J. Santos-Victor. Visual path following with a catadioptric panoramic camera. In *7th Int. Symp. on Intelligent Robotic Systems (SIRS'99)*, pages 139–147, Coimbra, Portugal, July 1999.

[11] R. M. Haralick and L. G. Shapiro. *Computer and Robot Vision (vol. 1)*. Addison-Wesley, 1992.

[12] A. Hicks and R. Bajcsy. Reflective surfaces as computational sensors. In *Workshop on Perception for Mobile Agents at CVPR'99*, Colorado, USA, June 1999.

[13] J. Hong, X. Tan, B. Pinette, R. Weiss, and E. Riseman. Image-based homing. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA'91)*, pages 620–625, 1991.

[14] D. Huttenlocher, R. Lilien, and C. Olsen. View-based recognition using an eigenspace approximation to the hausdorff measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(9):951–956, September 1999.

[15] H. Ishiguro and S. Tsuji. Image-based memory of environment. In *Proc. of Int. Conf. on Intelligent Robots and Systems (IROS'96)*, pages 634–639, 1996.

[16] S. D. Jones, C. Andersen, and J. L. Crowley. Appearance based processes for visual navigation. In *Proc. of the 5th Int. Conf. on Intelligent Robots and Systems (IROS'97)*, pages 551–557, July 1997.

[17] D. Kortenkamp and T. Weymouth. Topological mapping for mobile robots using a combination of sonar and visual sensing. In *AAAI'94*, July 1994.

[18] A. Kosaka and J. Pan. Purdue experiments in model-based vision for hallway navigation. In *Workshop on Vision for Robots at IROS'95*, Pittsburgh, PA, USA, 1995.

[19] J. Kosecka. Visually guided navigation. In *Proc. 4th Int. Symp. on Intelligent Robotic Systems (SIRS'96)*, Lisbon, Portugal, July 1996.

[20] D. Kriegman, E. Triendl, and T. Binford. Stereo vision and navigation in buildings for mobile robots. *IEEE Transactions on Robotics and Automation*, 5(6):792–802, December 1989.

[21] S. Maeda, Y. Kuno, and Y. Shirai. Active navigation vision based on eigenspace analysis. In *Proc. Int. Conf. on Intelligent Robots and Systems (IROS'97)*, pages 850–856, July 1997.

[22] H. Murase and S. K. Nayar. Visual learning and recognition of 3d objects from appearance. *Int. J. of Computer Vision*, 14(1):5–24, Jan. 1995.

[23] T. Pajdla and V. Hlavac. Zero phase representation of panoramic images for image based localization. In *8th Inter. Conf. on Computer Analysis of Images and Patterns CAIP'99*, 1999.

[24] W. Rucklidge. *Efficient Visual Recognition using the Hausdorff Distance*, volume 1173 of *Lecture Notes in Computer Science*. Springer-Verlag, 1996.

[25] J. Santos-Victor, R. Vassallo, and H. J. Schneebeli. Topological maps for visual navigation. In *1st Int. Conf. on Computer Vision Systems*, pages 21–36, Las Palmas, Canarias, January 1999.

[26] M. Srinivasan, K. Weber, and S. Venkatesh. *From living eyes to seeing machines*. Oxford University Press, 1997.

[27] M. A. Turk and A. P. Pentland. Face recognition using eigenfaces. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pages 586–591, 1991.

[28] R. Wehner and S. Wehner. Insect navigation: use of maps or ariadne's thread? *Ethology, Ecology, Evolution*, 2:27–48, 1990.

[29] N. Winters and J. Santos-Victor. Omni-directional visual navigation. In *Proc. of the 7th Int. Symp. on Intelligent Robotic Systems (SIRS'99)*, pages 109–118, Coimbra, Portugal, July 1999.