

Is structure needed for omnidirectional visual homing?*

Toon Goedemé*, Tinne Tuytelaars, and Luc Van Gool

VISICS - PSI - ESAT

University of Leuven

Belgium

*toon.goedeme@esat.kuleuven.ac.be

Dirk Vanhooydonck, Eric Demeester,

and Marnix Nuttin

PMA - Department of Mechanical Engineering

University of Leuven

Belgium

Abstract—Upcoming fast vision techniques for finding image correspondences enable reliable real-time visual homing, i.e. the guidance of a mobile robot from a arbitrary start pose towards a goal pose defined by an image taken there. Two approaches emerge in the field that differ in the fact that the structure of the scene is estimated or not. In this paper, we compare these two approaches for the general case and especially for our application, being automatic wheelchair navigation.

Index Terms—Computer vision, robot navigation, visual homing, omnidirectional images.

I. INTRODUCTION AND RELATED WORK

The goal of this research is to compare systems for mobile robot *visual homing*, i.e. the act of returning to a place by comparing the image currently viewed with an image taken at the goal position. Because their rich environment information content, we use omnidirectional images taken with a catadioptric image sensor mounted on the mobile robot. Note that this is the only sensor we use, no other environment information is acquired.

Homing is a term borrowed from biology, where it is usually used to describe the ability of various living organisms, such as insects, to return to their nest or to a food source after having traveled a long distance.

Many researchers have tried to imitate this behavior in mobile robots. Because the complexity working with images brings along, there have been many efforts to solve the navigation problem using non-visual sensors. Robots are usually equipped with odometry sensor systems that continuously measure the location of the robot with respect to some coordinate system. Unfortunately, the errors of these sensors are considerable and, most importantly, cumulative. For this reason, odometry is classically supplemented by information provided by ultrasound or laser sensors, mostly through the creation of a grid-based map of the environment [1]. Although many of these methods are quite successful, grid-based approaches must inevitably find a compromise between high space and time complexity and high accuracy regarding the number of grid cells. Moreover, robots based solely on proximity sensors get easily confused because the sensors often return similar readings at different locations. This problem can be patched by using Monte Carlo localization techniques [2].

Vision is, in comparison with these sensors, much more informative. We observe that many biological species, in particular flying animals, use mainly visual sensors for localization and homing.

Crucial in all visual homing methods is the selection of the landmarks. One must be able to find corresponding pixels between the present image and the target image. Franz [3] used a global optical-flow-like technique on the intensity string measured on the horizon circle. Also, Röfer *et al.* [4] computed a 1D optical flow using a modified Kohonen network. Much more descriptive and robust against occlusions is the technique of *local region matching*. In stead of looking at the image as a whole, local regions are defined around interest points in the images. The characterization of these local regions with descriptor vectors enables the regions to be compared across images. Because of the built-in invariance against photometric and geometric changes, correspondences can be found between images with different lighting and different viewpoints.

Many researchers proposed algorithms to do *local region matching*. The differences between approaches lie in the way in which interest points, local image regions and descriptor vectors are extracted. An early example is the work of Schmid and Mohr [5], where geometric invariance was still under image rotations only. Scaling was handled by using circular regions of several sizes. Lowe *et al.* [6] extended these ideas to real scale-invariance. More general affine invariance has been achieved in the work of Baumberg [7], that uses an iterative scheme and the combination of multiple scales, and in the more direct, constructive methods of Tuytelaars & Van Gool [8], [9], Matas *et al.* [10], and Mikolajczyk & Schmid [11]. Although these methods are capable to find very qualitative correspondences, most of them are too slow for use in a real-time mobile robot algorithm. That is why we spent efforts to speed this up, as explained later.

In omnidirectional visual homing implementations, one can choose between two main approaches. The first, inspired by the work of Cartwright and Collett [12], uses only the bearing (azimuth) angles of corresponding landmarks as input and does not use range or structure. Cartwright and Collett proposed, inspired by the visual homing behavior of insects such as bees and ants, the so-called 'snapshot' model. They suggest that insects fix the locations of landmarks surrounding a position by storing a snapshot image of the landmarks taken from that position. Their proposed algorithm consists of the construction of a home vector, computed as the average of landmark displacement vectors.

*This work is partially supported by the Inter-University Attraction Poles, Office of the Prime Minister, IUAP-AMS, the Flemish Institute for the Advancement of Science in Industry, IWT, and the Fund for Scientific Research Flanders (FWO-Vlaanderen, Belgium).

Franz *et al.* [3] analyzed the computational foundations of this method and derived its error and convergence properties. They conclude that every visual homing method based solely on bearing angles of landmarks, inevitably depends on basic assumptions such as equal landmark distances, isotropic landmark distribution or the availability of an external compass reference. Already, these assumptions indicate that for certain applications an alternative approach is necessary.

An other approach in visual homing, relies on the estimation of the range to the observed landmarks. In other words, the structure of the environment is represented in a 2D or 3D map. In this map, both present location and target location are estimated which enables the computation of a homing vector. Davison [13] developed a single camera SLAM method which estimates the ego-motion of the camera by building sparse probabilistic 3D maps with natural features. A motion model and loop-closing are implemented to reduce errors.

The remainder of this paper is organized as follows. In section II, we explain the innovative way landmark correspondences are established. Then, two methods to estimate the homing vector are described. Section III details a structure-less approach based on vector averaging, section IV describes the Kalman-filter based method with structure estimation. In section V, the experiments we have performed using both methods are compared and section VI draws a conclusion.

II. SOLVING THE CORRESPONDENCE PROBLEM

We have worked out a solution to the landmark correspondence problem, consisting of two steps. Because the initial and target images are taken relatively far from each other, initial correspondences are established using recently developed fast wide baseline matching algorithms, as explained in subsection II-A. While driving towards the target, an image sequence is recorded in which the formerly identified features are tracked, as described in subsection II-B.

A. Wide baseline feature correspondences

Although wide baseline local features are common in computer vision, only recently, a class of *fast* wide baseline local features have appeared. We use the combination of two different kinds of these features, namely a rotation reduced and color enhanced form of David Lowe's *SIFT features* [6], and the *invariant column segments* we developed in previous work [14], [15].

1) *Rotation reduced and color enhanced SIFT*: David Lowe presented the *Scale Invariant Feature Transform* [6], which finds interest points around local peaks in a series of difference-of-Gaussian (DoG) images. A dominant gradient orientation and scale factor define an image patch around each interest point so that a local image descriptor can be found as the histogram of the gradient directions of the normalized image patch around the interest point. SIFT features are invariant to rotation and scaling, and robust to other transformations.

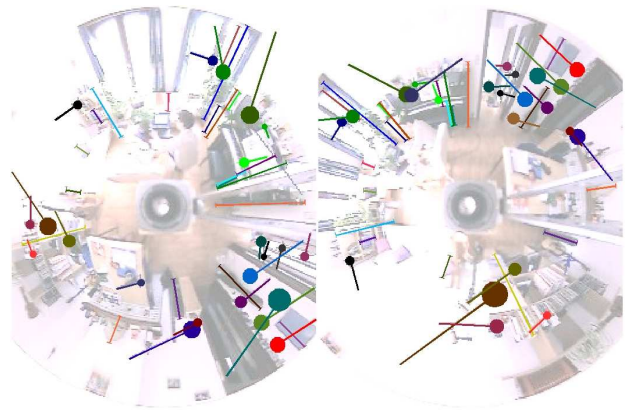


Fig. 1. A pair of omnidirectional images, superimposed with color-coded corresponding column segments (radial lines) and SIFT features (circles with tail).

A reduced form of these SIFT features for use on mobile robots is proposed by Ledwich and Williams [16]. They used the fact that rotational invariance is not needed for a camera fixed on a mobile robot moving on one plane. Elimination of the rotational normalization and rotational part of the descriptor yields a somewhat less complex feature extraction and more robust feature matching performance.

Because the original SIFT algorithm works on grayscale images, some mismatches occur at similar objects in different colors. That is why we propose an outlier filtering stage based on a color descriptor of the feature patch based on global color moments, introduced by Mindru *et al.* [17]. We chose the following three color descriptors: C_{RB} , C_{RG} and C_{GB} , with

$$C_{PQ} = \frac{\int PQ dx \int dx}{\int P dx \int Q dx}, \quad (1)$$

where R , G and B are the red, green, and blue color components along the column segment, centralized around their means. After matching, the correspondences with Euclidean distance between the color description vectors above a fixed threshold are discarded.

Between the image pair in fig. 1 the original SIFT algorithm finds 13 correct matches. Using our rotation reduced and color enhanced algorithm, the matching threshold can be raised so that up to 25 correct matches are found without including erroneous ones.

2) *Invariant column segments*: In earlier work [14], we developed wide baseline features which are specially suited for mobile robot navigation. Taking advantage of the movement constraints of a fixed camera on a robot moving on a plane, a very simple and fast algorithm can be carried out. The (dewarped) image is scanned columnwise and column segment features are defined between two local maxima of the image gradient. Each column segment is described by an 11-element vector containing information on geometry, color and intensity.

Fig. 1 shows the matching results on a pair of omnidirectional images. As seen in these examples, the SIFT features and the column segments are complementary, which pleads for the combined use of the two.

B. Feature tracking

The corresponding features between the first image and the target image, found in the previous step, also have to be found in the incoming images during driving. This can be done very reliably performing every time wide baseline matching with first or target image, or both. Although recent methods are relatively fast (about 0.8s for a pair of 640×480 images, see [14]), this is still too time-consuming for a driving robot.

Because the incoming images are part of a smooth continuous sequence, a better solution is *tracking*. In the image sequence, visual features move only a little from one image to the next, which enables to find the new feature position in a small search space.

A widely used tracker is the KLT tracker of Kanade, Lucas, Shi, and Tomasi [18], [19]. KLT starts by identifying interest points (corners), which then are tracked in a series of images. The basic principle of KLT is that the definition of corners to be tracked is exactly the one that guarantees optimal tracking. A point is selected if the matrix

$$\begin{bmatrix} g_x^2 & g_x g_y \\ g_x g_y & g_y^2 \end{bmatrix}, \quad (2)$$

containing the partial derivatives g_x and g_y of the image intensity function over an $N \times N$ neighborhood, has large eigenvalues. Tracking is then based on a Newton-Raphson style minimization procedure using a purely translational model. This algorithm works surprisingly fast: we were able to track 100 feature points at 10 frames per second in 320×240 images on a 800 MHz laptop.

Because the well trackable points are not necessarily coinciding with the center points of the wide baseline features to be tracked, the best trackable point in a small window around such a center point is selected. In the assumption of local planarity we can always locate the point in the relative reference system the wide baseline feature offers.

III. STRUCTURE-LESS HOMING METHOD

The method described in the previous section yields a set of reliable and accurate corresponding points between each image during the visual homing procedure and the target image. This is the dreamed starting data for each correspondence-based visual servoing algorithm.

In the wealth of algorithms derived from the snapshot model of Cartwright and Collet [12], we chose the implementation of Argyros *et al.* In his paper [20], Argyros stresses the fact that his method achieves robot navigation without computing any explicit range information at all, although silently he assumes an isotropic landmark distribution.

At each time instant, a homing vector is computed based on bearing angle measurements of corresponding landmarks in present and goal position. Fig. 2 illustrates the method. Let P_i denote the bearing angle of the i 'th feature in the present image and G_i the bearing angle of the feature in the goal image. Two features i and j define two

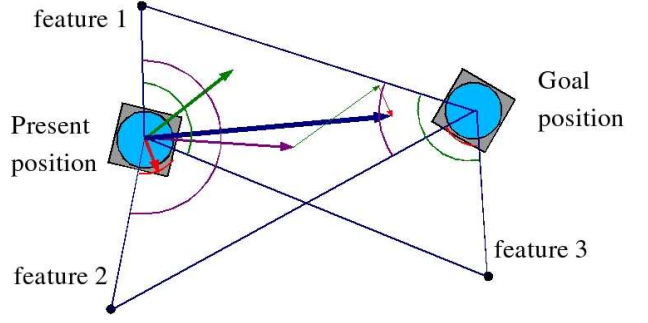


Fig. 2. Estimation of the homing vector between two camera poses given at least three corresponding visual feature pairs.

sectors in the images, with angular sizes $P_{ij} = |P_j - P_i|$ and $G_{ij} = |G_j - G_i|$, respectively. Since features can be corresponded, sectors can also be corresponded. For each corresponding sector pair, the difference in its angular size in the two images is examined. If the difference is positive, a vector is defined parallel to the bisector of the sector in the present image, and with a magnitude proportional to the difference $|G_{ij} - P_{ij}|$. If the difference is negative, then the vector has opposite direction. In fact, each of these vectors is targeted towards the direction of the motion that is required in order to equalize the angular extend of the sector between the two views. The vector sum of all contributions from all sectors defines the instantaneous homing vector.

IV. HOMING METHOD WITH STRUCTURE ESTIMATION

Unlike the former method, the method we propose in this section builds a local map containing the landmark positions. At each time instant, both the present robot location as the target location are estimated therein, so that a homing vector can be calculated.

After initialization of the feature correspondences, the algorithm computes a robust estimate of the homing vector and drafts a local map, containing the feature world positions. This is described in subsection IV-A. Later, during the movement, as feature measurements are coming in, the map is refined and the position of the present and goal images is updated, see subsection IV-B.

A. Homing vector and local map initialization step

From a set of feature correspondences an initial homing vector can be estimated. In spite of our efforts, still feature mismatches can occur, for instance in the presence of multiple identical environment objects. Because the homing vector estimation is the basis for navigation and later corrections, a robust estimation technique must be used.

We found the solution in a *random sample consensus* (RANSAC) [21] estimation scheme. Out of a randomly chosen set of two correspondences, the target pose relative to the present pose can be computed using goniometrics. Fig. 3 illustrates this.

The angles labeled with α in the side view (top right) can be extracted from the image coordinates of the features. In order to do this, a 5 degree polynomial is fitted to the data of a calibration image like the one shown bottom right in

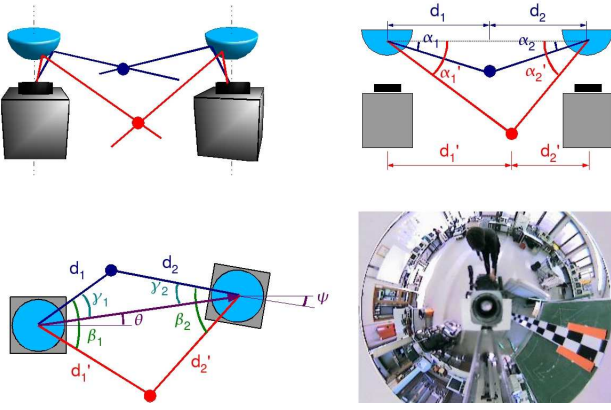


Fig. 3. Estimation of the homing vector between two camera poses given two corresponding visual features. The red and blue lines indicate the light rays towards the two features, respectively. Top left: perspective view, top right: side view (folded open so that the distances d are not shortened), bottom left: top view, bottom right: image used for calibration of angles α .

fig. 3. It is clear that $d_1 \tan \alpha_1 = d_2 \tan \alpha_2$. In other words, we do not know the distances d themselves, but the ratio between them is known.

Now observe the top view, shown bottom left. Using the latter and the fact that the bearing difference angles β can be measured in the image, the quadrangle camera1-feature1-camera2-feature2 can be computed up to an unknown scalefactor.

This yields a homing vector (R, θ, ψ) , with (R, θ) the polar coordinates of the target position relative to the present position and ψ the robot orientation change between the two poses. Unfortunately, the vision scale ambiguity forces us to set R arbitrarily at 1.

A second step in the RANSAC algorithm is counting the inlier rate as a measure for the support of the estimated homing vector. In order to do so, for each correspondence the ratio $\frac{d_1}{d_2}$ is computed both as $\frac{\sin \gamma_2}{\sin \gamma_1}$ and $\frac{\tan \alpha_2}{\tan \alpha_1}$. If both outcomes differ too much, the correspondence is considered an outlier.

This calculation is repeated a sufficient number of times (200 in our experiments) on different random samples, and the homing vector with the highest support is kept.

In order to start up the succession of map and location updating iterations, an estimate of the local map must be made. In our approach the local map is a 2D representation of the world, centered at the starting position of the visual homing operation. The world position of every corresponding visual feature used in the previous step must be represented in this map. Because the homing vector and the feature bearing angles seen from both viewpoints are known, the latter is computed easily by triangulation. Fig. 4 gives an example.

B. Homing vector and local map update step

When estimates of the homing vector and local map are found, the robot is put into motion in the direction of that homing vector. We rely on a lower-level collision detection and obstacle avoidance algorithm to do this safely. During this drive, images are taken giving information to update

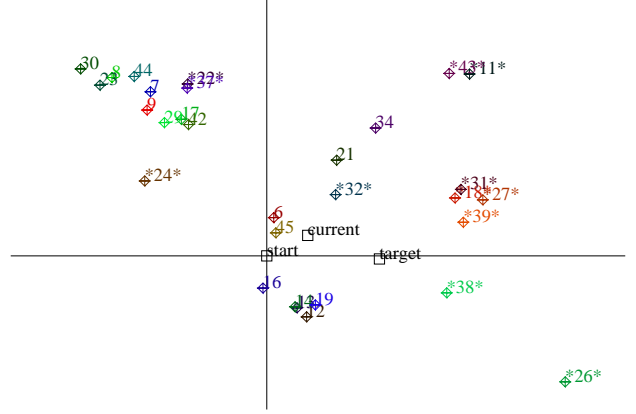


Fig. 4. Example of a feature map built up by our algorithm. Black squares show start, current and target robot positions. Colored diamonds show feature positions, labeled with their number. Stars in the feature label denote a feature that is lost during tracking.

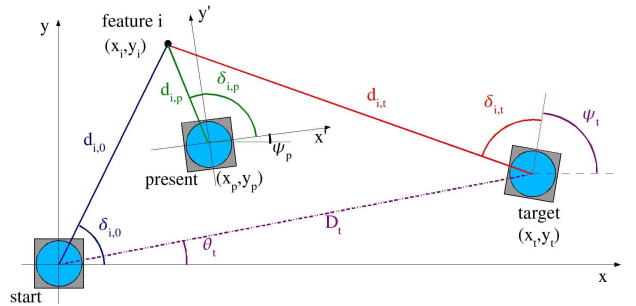


Fig. 5. Symbol definition for the Kalman filters.

the local map and the location of the robot in that local map, also known as *Simultaneous Localization And Mapping* or *SLAM*. Each time the target position can be updated too, so that an updated homing vector can be derived. We implemented these updating operations as extended Kalman filters (EKF), using the image positions of the tracked visual features as inputs.

When close enough to one target, the movement towards the next target image is started. This yields a smooth trajectory along a sparsely defined visual path.

1) *Local map updating*: During movement, each time an image is taken new information of the environment is achieved and the local map of features can be corrected. Also, the position of the robot in that local map has changed and has to be corrected. For this SLAM-task, we implemented an Extended version of the Kalman Filter [22] (EKF), inspired on the work of Azarbayejani and Pentland [23] and similar to Kim and Chung [24], who adapted this algorithm for use with omnidirectional images.

Fig. 5 explains the various symbols used. The state vector we continuously update contains the present robot pose, the robot speed, and the polar feature coordinates:

$$\mathbf{x}_p = (x_p, y_p, \psi_p, v_p, d_{1,0}, \delta_{1,0}, d_{2,0}, \delta_{2,0}, \dots, d_{N,0}, \delta_{N,0}). \quad (3)$$

In the movement equation we express the expectation that

the robot drives forward with his present velocity:

$$\hat{x}_p(k) = x_p(k-1) + v_p(k-1)\Delta t \cos \psi_p \quad (4)$$

$$\hat{y}_p(k) = y_p(k-1) + v_p(k-1)\Delta t \sin \psi_p. \quad (5)$$

Each iteration, a new measurement vector is acquired containing the feature bearings as seen from the new pose:

$$\mathbf{z}_p(k) = (\delta_{1,p}(k), \delta_{2,p}(k), \dots, \delta_{N,p}(k)). \quad (6)$$

The measurement equation \mathbf{H}_p gives the estimate of these observations based on the present state vector. The i 'th element of this vector is the function

$$\hat{z}_{p,i}(k) = H_{p,i}(\hat{x}_p(k)) = \arctan\left(\frac{y'_i}{x'_i}\right), \quad (7)$$

with (x'_i, y'_i) the coordinates of the i 'th feature relative to the present robot pose

$$\begin{bmatrix} x'_i \\ y'_i \end{bmatrix} = \begin{bmatrix} \cos \psi_p & -\sin \psi_p \\ \sin \psi_p & \cos \psi_p \end{bmatrix} \begin{bmatrix} d_{i,0} \cos \delta_{i,0} \\ d_{i,0} \sin \delta_{i,0} \end{bmatrix} - \begin{bmatrix} \hat{x}_p \\ \hat{y}_p \end{bmatrix}. \quad (8)$$

With this model a Extended Kalman Filter is set up. The choice for polar coordinates facilitates the computation of the necessary Jacobians of motion and measurement equation. The state vector is initialized according to the estimated local map of subsection IV-A. We also set the initial variances on δ_i to zero to avoid instabilities, like in [23]. The variance on one of the d_i 's is set to zero, to fix the scale. When during tracking features are lost, the corresponding elements of the state vector and the corresponding rows and columns of the covariance matrices are deleted, as shown in [24].

2) *Homing vector updating*: The former EKF filter enables the construction of a local map with increasing accuracy and the present localization of the robot therein. However, only an estimate of the position where the target image is taken is known. A similar EKF procedure can be used to refine the target location as more accurate map information becomes available. The homing vector used to steer the robot is then the relative vector between the present location of the robot and that target position. As every distance, also this vector is only known up to a scale factor. But the relative length w.r.t. the original homing vector (with length 1) gives information for the robot how far to drive still.

We construct a second extended Kalman filter for the target localization. The state vector is the target pose: $\mathbf{x}_t = (D_t, \theta_t, \psi_t)$. We assume the target to be static, so the movement equation is simply $\hat{\mathbf{x}}_t(k) = \mathbf{I} \cdot \mathbf{x}_t(k-1)$. The measurements are the feature bearing angles as seen from the target, $\mathbf{z}_t = (\delta_{1,t}, \delta_{2,t}, \dots, \delta_{N,t})$. These measurements are predicted using the measurement equation \mathbf{H}_t :

$$\hat{z}_{t,i}(k) = H_{t,i}(\hat{\mathbf{x}}_t(k)) = \arctan\left(\frac{y_i - y_t}{x_i - x_t}\right) - \psi_t, \quad (9)$$

Also in this case the computation of the Jacobians pose no problems. For the initialization of this EKF, the estimated data from the initialization step (section IV-A) are used.



Fig. 6. Target position (left) and start position (right) of the wheel chair, with superimposed projected motion (black: without structure, light green: using structure).

V. EXPERIMENTAL RESULTS

We have implemented the proposed algorithm, using a modified electric wheel chair as test platform, described in V-A. The results of the tests are detailed in V-B.

A. Test platform

Our wheelchair "Sharioto" (depicted in fig. 6) is a standard electric wheelchair that has been equipped with an omnidirectional vision sensor (consisting of a Sony firewire color camera and a hyperbolic mirror). The image processing is performed on a 800 MHz laptop. As additional sensors for obstacle detection, four ultrasound sensors without dead zone and a Lidar are added. A second laptop with a 418 MHz processor reads these sensors, receives visual homing vector commands, computes the necessary manoeuvres, and drives the motors via a CAN-bus. More information can also be found in [25] and [26].

B. Results

We have done extensive tests of the two described methods for visual homing in a natural environment. As seen in the images, no special visual features were added to the natural scene. First, a target image was taken at a certain position which was marked on the floor for future reference (left in fig. 6). Then, the wheel chair is placed at a distance of 1 to 2 metres from this target and the homing procedure is performed. A typical result for the trajectories is shown in fig. 7, and superimposed in fig. 6.

The graph in fig. 8 shows the average computation times for one home vector update cycle. Although the methods have the image acquisition and KLT feature tracking in common, the remaining time for the actual homing vector computation varies greatly: 0.44 ms for the method without structure estimation (barely visible on the graph), and 71 ms for the method with structure estimation. The time needed for initialization of the movement, including wide baseline matching, averages to 858 ms.

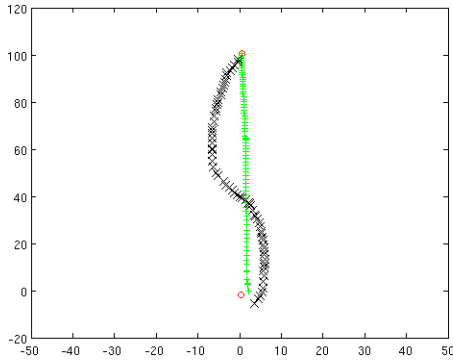


Fig. 7. Motion path (black: without structure, light green: using structure).

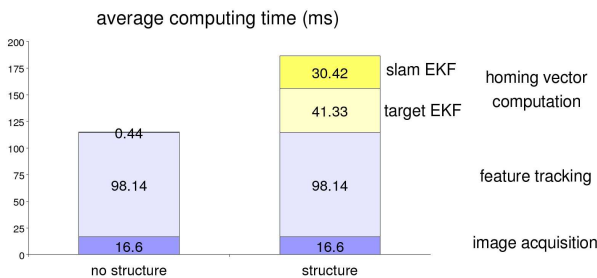


Fig. 8. Comparison of the computation times of the two methods.

VI. CONCLUSION

In this work, we have compared omnidirectional visual homing methods, based on visual correspondences computed by the novel method we developed. First, a method that does not rely on the estimation of the structure of the scene, based on [20], was evaluated. These results were compared with the results the method we developed generates. Our method does estimate a sparse 2D map of the environment.

Both methods were implemented on a electric wheelchair platform. We can conclude that the path our method gives is more accurate and smooth. However, the computational load the structure-less method posed was substantially lower, which enables a higher updating frequency.

With our correspondence solution, there is no guarantee of a isotropic correspondence distribution around the horizon circle. This fact is the reason for the relatively small homeward component of the homing vector of the structure-less method, which assumes such a distribution.

ACKNOWLEDGMENT

T.G. thanks Gerolf Vanacker for the wheel chair implementation work.

REFERENCES

- [1] S. Thrun, "Learning Metric-Topological Maps for Indoor Mobile Robot Navigation," *AI Journal*, 99(1), pp. 21-71, 1999.
- [2] S. Thrun, D. Fox, W. Burgard, and F. Dellaert, "Robust Monte Carlo Localization for Mobile Robots," *AI Journal*, vol. 128, nb. 1-2, pp. 99-141, 2001.

- [3] M. Franz, B. Schölkopf, H. Mallot, and H. Bülthoff, "Where Did I Take that Snapshot? Scene-based homing by image matching," *Biological Cybernetics*, 79, pp. 191-202, 1998.
- [4] T. Röfer, "Controlling a Wheelchair with Image-based Homing," In *Spatial Reasoning in Mobile Robots and Animals*, AISB-97 Workshop, Manchester University, 1997.
- [5] C. Schmid, R. Mohr, C. Bauckhage, "Local Grey-value Invariants for Image Retrieval," *International Journal on Pattern Analysis and Machine Intelligence*, Vol. 19, no. 5, pp. 872-877, 1997.
- [6] D. Lowe, "Object Recognition from Local Scale-Invariant Features," *International Conference on Computer Vision*, pp. 1150-1157, 1999.
- [7] A. Baumberg, "Reliable Feature Matching across Widely Separated Views," *Conference on Computer Vision and Pattern Recognition*, Hilton Head, South Carolina, pp. 774-781, 2000.
- [8] T. Tuytelaars, L. Van Gool, L. D'haene, and R. Koch, "Matching of Affinely Invariant Regions for Visual Servoing," *ICRA*, pp. 1601-1606, 1999.
- [9] T. Tuytelaars and L. Van Gool, "Wide Baseline Stereo based on Local, Affinely Invariant Regions," *British Machine Vision Conference*, Bristol, UK, pp. 412-422, 2000.
- [10] J. Matas, O. Chum, M. Urban and T. Pajdla, "Robust Wide Baseline Stereo from Maximally Stable Extremal Regions," *British Machine Vision Conference*, Cardiff, Wales, pp. 384-396, 2002.
- [11] K. Mikolajczyk and C. Schmid, "An Affine Invariant Interest Point Detector," *ECCV*, vol. 1, 128-142, 2002.
- [12] B. Cartwright, T. Collett, "Landmark Maps for Honeybees," *Biol. Cybernetics*, 57, pp. 85-93, 1987.
- [13] A. Davison, "Real-time Simultaneous Localisation and Mapping With a Single Camera," *International Conference on Computer Vision ICCV'03*, Nice, France, 2003.
- [14] T. Goedemé, T. Tuytelaars, and L. Van Gool, "Fast Wide Baseline Matching with Constrained Camera Position," *Computer Vision and Pattern Recognition*, Washington, DC, pp. 24-29, 2004.
- [15] T. Goedemé, M. Nuttin, T. Tuytelaars, and L. Van Gool, "Markerless Computer Vision Based Localization using Automatically Generated Topological Maps," *European Navigation Conference GNSS*, Rotterdam, 2004.
- [16] L. Ledwich and S. Williams, "Reduced SIFT Features For Image Retrieval and Indoor Localisation," *Australasian Conference on Robotics and Automation*, Canberra, 2004.
- [17] F. Mindru, T. Moons, and L. Van Gool, "Recognizing Color Patterns Irrespective of Viewpoint and Illumination," *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 368-373, 1999.
- [18] B. Lucas and T. Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision," *Intl. Joint Conf. on Artificial Intelligence*, pp. 674-679, 1981.
- [19] J. Shi and C. Tomasi, "Good Features to Track," *IEEE Conf. on Computer Vision and Pattern Recognition*, Seattle, pp. 593-600, 1994.
- [20] A. Argyros, K. Bekris, and S. Orphanoudakis, "Robot Homing based on Corner Tracking in a Sequence of Panoramic Images", *Conference on Computer Vision and Pattern Recognition*, vol. 2, p. 3, Kauai, Hawaii, 2001.
- [21] M. Fischler, R. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis," *Comm. of the ACM*, Vol 24, pp 381-395, 1981.
- [22] R. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Trans. of the ASME-Journal of Basic Engineering*, vol. 82, Series D, pp. 35-45, 1960.
- [23] A. Azarbayejani and A. P. Pentland, "Recursive Estimation of Motion, Structure, and Focal Length," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, no. 6, pp. 562-575, June 1995.
- [24] J.-H. Kim and M. J. Chung, "SLAM with Omni-directional Stereo Vision Sensor," *proc. of IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, Las Vegas, Nevada, 2003.
- [25] M. Nuttin, E. Demeester, D. Vanhooydonck, and H. Van Brussel, "Shared Autonomy for Wheel Chair Control: Attempts to assess the user's autonomy," in *Autonome Mobile Systeme AMS2001*, Stuttgart, pp. 127-133, 2001.
- [26] E. Demeester, M. Nuttin, D. Vanhooydonck, and H. Van Brussel, "Fine Motion Planning for Shared Wheelchair Control: Requirements and Preliminary Experiments," in *proc. of the Intl. Conf. on Advanced Robotics (ICAR)*, Coimbra, pp. 1278-1283, 2003.
- [27] T. Goedemé, M. Nuttin, T. Tuytelaars, and L. Van Gool, "Vision Based Intelligent Wheelchair Control: The Role of Vision and Inertial Sensing in Topological Navigation," *Journal of Robotic Systems*, 21(2), pp. 85-94, 2004.