

Visual map-less navigation based on homographies *

J.J. Guerrero, R. Martinez-Cantin, C. Sagüés
Dpto. de Informática e Ingeniería de Sistemas /
Instituto de Investigación en Ingeniería de Aragón
Universidad de Zaragoza
C) María de Luna num. 1, E-50018 Zaragoza, Spain
jguerrer, rmcantin, csagues@unizar.es

June 3, 2005

Abstract

We introduce a method for autonomous robot navigation based on homographies computed between current image and images taken in a previous teaching phase with a monocular vision system. The features used to estimate the homography are vertical lines automatically extracted and matched. From homography, the underlying motion correction between the reference path and the current robot location is computed. The proposed method, which uses a sole calibration parameter, has turned out to be specially useful to correct heading and lateral displacement, which are critical in systems based on odometry. We have tested the proposal in simulation, and with real images. Besides, the visual system has been integrated into an autonomous wheelchair for handicapped, working in real time with robustness.

Key words: Visual navigation, Homographies, Robust matching, Robot homing, Wheelchair for handicapped.

*This work was supported by project DPI2003-07986.

1 Introduction

We would like to start with a simple analogy. Let us suppose we are in Paris and we need to return to the "Rue de l'Université" where our hotel is. First, we could have memorized the number of steps walked from the hotel and we could go back trying to count the same number of steps (dead-reckoning navigation). We could buy a map and use it to reach the goal (map-based navigation). This solution requires that somebody have named the streets and have drawn the map. We could also explore Paris, simultaneously drawing our own map (map-building navigation), but this is time costly. Finally, we can look around trying to find the Eiffel Tower, and then we can try to approach there keeping the top of the tower in our field of view (map-less navigation). The goal is to get the tower because we know that the "Rue de l'Université" starts there.

Normally, autonomous vehicle navigation uses some of these solutions to track a safe and efficient trajectory towards the programmed goal. The commonest and cheapest method is the first one, especially in wheeled vehicles that usually include an odometry system. However, this solution presents several problems (slipping, mechanical drifts, ...) that produce a cumulative error which is unacceptable in long term navigation. So, an additional perception system is mandatory. Vision is perhaps the most broadly researched perception system.

Map-less navigation is barely new compared with the map-based solutions, but several works using vision systems have been developed in several directions in the last years [1]. So, there are works where optical flow is used imitating the visual behavior of bees [2]. One of the most known methods of visual map-less navigation is the view-based navigation system where a representation of the route with a sequence of images is built. In that case the motion is obtained from a correlation based matching of the current image and the reference image of the memorized sequence [3]. Our system can be classified in this group because our vehicle can autonomously navigate without prepared landmarks or complex map-building systems, but using specified target positions with images memorized in the teaching phase.

In this paper, we present a map-less solution based on homographies to compute relative displacements between current camera view and reference images captured in known locations. It has been developed for indoor navigation with low computational and memory cost. Processing the current image and the stored image, a projective transformation is obtained to move, in

the playback phase, towards the goal. The geometric features extracted and matched are lines which, have some advantages compared to points [4], specially in man made environments. Other authors [5] also use vertical lines in a teaching and playback approach to correct motion, but using a calibrated trinocular vision system.

Although global motion, translation and rotation, could be obtained from the computed homography, we approach the rotation from one parameter, which gives the suitable heading in a more robust way. Initially, it is used to correct orientation errors that are the most critical in robot navigation. However, this computed heading has proved to correct lateral translation when camera looks at the direction of advance of the robot. In other vision based approaches [6], translation is corrected in two steps because the estimation of the scale factor is needed.

Some advantages of our approach are the simplicity to correct heading, because it only needs a sole calibration parameter of the wheel-eye system, and its robustness to inaccuracies. Additionally, we do not store raw images because geometric information (lines) are used. Besides that, extracting and matching geometric information from images is not actually time costly and geometric based approaches are less sensitive to noise or illumination changes than others.

In other works, corresponding points have been used for robot navigation using coplanar constraints [7], but normally they have been considered to compute the fundamental matrix [6], which is a more general model. However, the fundamental matrix is ill conditioned with planar scenes, which are quite usual in man made environments [8],[9]. Besides that, many times, in this application small baseline images with high disparity due to rotation are computed, where fundamental matrix also can result an inappropriate model to compute robot motion.

The proposed correction system has been integrated into the navigation system using our efficient and robust technique to extract and match line features, computing simultaneously homographies [10]. It has been tested indoors in a long term navigation making real time corrections without drift problems. Additionally, it has also been tested that using a sole reference image, the autonomous system may cross doors, which is a difficult problem for handicapped driving without help.

The first part of the paper presents an overview of robust homography estimation from image features, including techniques to compute motion from homography. Then, we prove the techniques using computer generated and

real data. Finally, we have tested the complete process in a real-time implementation which works on a wheelchair for handicapped.

2 Matching lines and computing homographies

As explained, in our proposal, motion information is obtained from the comparison between features in previously recorded images and the features in images taken at each position reached by the wheelchair when it moves to the goal. An homography relates points or lines in one image belonging to a plane of the scene with points or lines in the other image. This model is exact for planar scenes and it is a good approximation in general scenes where translation between images is small with respect to scene depth. This turns out useful in robotic applications.

Our approach takes lines in the image as key features, because they are plentiful in man made environments. The straight lines have a simple mathematical representation, they can be extracted more accurately than points, being also easier to match and they can be used in cases where there are partial occlusions. After extracting the lines, automatic computation of correspondences and homographies is carried out as previously presented [10]. Thus, initially the extracted lines are matched to the weighted nearest neighbor using brightness-based and geometric-based image parameters. From them, robust homographies are computed.

In most cases, and specially when vehicle moves in man made environments, the motion is on a plane and vertical lines give enough information to carry out homing. With vertical lines, only the x coordinate is relevant and the computation problem is simplified. As the homography now has four parameters up to a scale factor, and each vertical line gives one equation, three matches are enough to compute the homography. In this case, the parameter referred to the orientation of the lines is discarded, but in practice the lines are grouped into two incompatible subsets, those having 90 degrees of orientation and those with 270 degrees of orientation, in function of the dark side of the edge.

The proposed matching is similar to the presented for lines in all directions [10]. Here, we use an homogeneous representation where a vertical line can be projectively represented as $(x, 1)$. Therefore for each corresponding vertical

line having x_{m1} and x_{m2} coordinates in both images we have

$$\begin{pmatrix} \lambda & x_{m2} \\ & \lambda \end{pmatrix} = \mathbf{H}_{21} \begin{pmatrix} x_{m1} \\ 1 \end{pmatrix} = \begin{pmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{pmatrix} \begin{pmatrix} x_{m1} \\ 1 \end{pmatrix},$$

which provides one equation to solve \mathbf{H}_{21}

$$\begin{pmatrix} x_{m1} & 1 & -x_{m1}x_{m2} & -x_{m2} \end{pmatrix} \begin{pmatrix} h_{11} \\ h_{12} \\ h_{21} \\ h_{22} \end{pmatrix} = 0.$$

With the coordinates of at least, three vertical lines, we can construct a 3×4 matrix \mathbf{M} . The homography solution corresponds to the eigenvector associated to the least eigenvalue of the $\mathbf{M}^T \mathbf{M}$ matrix and it can be solved by singular value decomposition of \mathbf{M} matrix.

As usually we have more than three matches, and an estimation method can be used to process all of them, getting better results. The least squares method assumes that all the measures can be interpreted with the same model, which makes it to be very sensitive to out of norm data. Robust estimation tries to avoid the outliers in the computation of the estimate. From the existing robust estimation methods [11], we have chosen the least median of squares method. This method makes a search in the space of solutions obtained from subsets of minimum number of matches (Algorithm 1).

Algorithm 1 Robust estimation of homography

- 1: A Monte-Carlo technique is used to randomly select m subsets of 3 features.
 - 2: For each subset S , we compute a solution in closed form \mathbf{H}_{21} .
 - 3: For each solution \mathbf{H}_{21} , the median M_S of the squares of the residue with respect to all the matches is computed.
 - 4: We store the solution \mathbf{H}_{21} which gives the minimum median M_S .
-

A selection of m subsets is good if at least in one subset the 3 matches are good. If P_{ns} is the probability that a putative match is not spurious and P_f is the probability of not reaching to a good solution, the number of subsets to consider can be computed as: $m = \frac{\log P_f}{\log(1-P_{ns}^3)}$. For example, if we accept a probability $P_f = 0.01$ of fail, with an estimation of the probability

$P_{ns} = 70\%$ of good matches, the number of subsets m should be 11. But a 10 times reduction of the probability of fail ($P_f = 0.001$) can be considered increasing the computational cost less than 0.6 times, if 17 subsets are taken.

Once the solution has been obtained, the outliers can be selected from those of higher residue to compute without them a better solution. The threshold to select the outliers is taken from the standard deviation of the residue [12], estimated as $\hat{\sigma} = 1.4826 [1 + 5/(n - 3)] \sqrt{M_S}$, n being the number of matches. Assuming that the measurement error is Gaussian with zero mean and standard deviation σ , then the square of the residues follows a χ_1^2 distribution with 1 degree of freedom. Taking for example, a 5% probability of rejecting a line matching being inlier, the threshold will be fixed to $3.84 \hat{\sigma}^2$.

3 Motion from homography

Motion information is obtained from the previously recorded image and the current image, taken both with the same camera. As told, planar motion is to be computed and 2D homography is enough. From that homography, complete motion (rotation and translation up to a scale factor) is computed, but correction of rotation approached from homography parameters as explained below is enough robust to make the robot navigate properly.

3.1 Complete motion algorithm

Let us suppose two images whose projection matrixes in a common reference system are $\mathbf{P}_1 = \mathbf{K}[\mathbf{I}|\mathbf{0}]$ and $\mathbf{P}_2 = \mathbf{K}[\mathbf{R}|\mathbf{t}]$ being \mathbf{K} the internal calibration matrix, \mathbf{R} the camera rotation and \mathbf{t} the translation to move from second to first camera location. The homography \mathbf{H}_{21} can be related to motion in such a way that,

$$\mathbf{H}_{21} = \mathbf{K} \left(\mathbf{R} - \frac{\mathbf{t} \mathbf{n}^T}{d} \right) \mathbf{K}^{-1} \quad (1)$$

being \mathbf{n} the normal to the scene plane and d its depth.

The camera motion and the planar structure can be computed from \mathbf{H}_{21} when the camera is calibrated. From here, two solutions (with a scale factor for \mathbf{t}) can be obtained [13]. The main steps of our 2D adaptation of this algorithm is summarized (Algorithm 2).

Algorithm 2 Complete motion algorithm

- 1: Compute a calibrated homography $\mathbf{H} = \mathbf{K}^{-1} \mathbf{H}_{21} \mathbf{K}$
 - 2: Compute the singular value decomposition of matrix \mathbf{H} , as $\mathbf{H} = \mathbf{U} \mathbf{S} \mathbf{V}^T$
 - 3: Let be $\mathbf{S}^T \mathbf{S} = \text{diag}(\lambda_1, \lambda_2)$, and $\alpha = \sqrt{\frac{\lambda_1-1}{\lambda_2-\lambda_1}}$, $\beta = \sqrt{\frac{1-\lambda_1}{\lambda_2-\lambda_1}}$
 - 4: Writing $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2]$, compute $\mathbf{v}_v = \alpha \mathbf{v}_1 + \beta \mathbf{v}_2$.
 - 5: Compute rotation matrix as $\mathbf{R} = [\mathbf{H} \mathbf{v}_v, \mathbf{P}_\perp \mathbf{H} \mathbf{v}_v][\mathbf{v}_v, \mathbf{P}_\perp \mathbf{v}_v]^T$ being $\mathbf{P}_\perp = [0, -1; 0, 1]$
 - 6: Compute translation up to a scale factor as $\mathbf{t} = \mathbf{R} \mathbf{n} - \mathbf{H} \mathbf{n}$ being $\mathbf{n} = \mathbf{P}_\perp \mathbf{v}_v$
 - 7: The second solution for \mathbf{R} and \mathbf{t} can be obtained by making $\beta = -\beta$.
 - 8: If $\lambda_1 = 1$ there is a sole solution, being the camera translation perpendicular to the plane ($\mathbf{t} \parallel \mathbf{R} \mathbf{n}$) and coming nearer the plane. If $\lambda_2 = 1$ there is also a sole solution, but now the camera gets away from the plane. Finally, if $\lambda_1 = \lambda_2 = 1$ report the sole solution $\mathbf{t} = 0$, and $\mathbf{R} = \mathbf{H}$
-

This algorithm provides in a general case two solutions, being a problem in practice to select the correct one. When the camera translation is perpendicular to the scene plane (for example when robot goes to cross a door) there is a sole solution.

3.2 Single-parameter motion correction

The relative motion between the first camera location and the planar scene, can be written in function of d distance and the orientation φ of the scene with respect to the camera location (Fig. 1). Similarly the transformation between camera locations can be written in function of the translation t_x, t_z and rotation θ between both.

From there, the homography can be expressed as,

$$\mathbf{H}_{21} = \begin{bmatrix} d C\theta + S\varphi t_x & f (d S\theta + t_x C\varphi) \\ \frac{1}{f}(S\varphi t_z - d S\theta) & C\varphi t_z + d C\theta \end{bmatrix} \quad (2)$$

being f the internal camera parameter, that includes the focal distance and the image size. Here, it has been supposed a centered image where the camera reference system is on the principal point.

If the effect of camera translation is shorter than that of the camera rotation, then the influence of the scene disappears and the camera rotation can be obtained from two elements of the homography \mathbf{H}_{21} , once they have

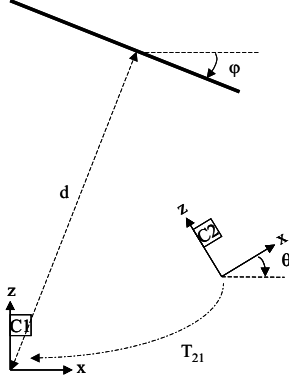


Figure 1: Motion between cameras c1 and c2 and planar scene.

been normalized, as

$$\mu = \frac{h_{12}}{h_{22}} = f \tan(\theta) \quad ; \quad t = \frac{h_{21}}{h_{22}} = -\tan(\theta)/f$$

In this way, rotation can be computed in two ways from μ and t , being both proportional to $\tan(\theta)$ through an internal parameter (f) which may be easily obtained with a previous calibration.

Additionally in this case, camera rotation (θ) can also be computed from homography eigenvalues that are $e^{j\theta}$ and $e^{-j\theta}$. In this way, the camera calibration is not needed, provided that the same camera is used to capture both images. Rotation direction, however, is not obtained.

As said, rotation is the most important information to make a robot to correct its trajectory. Relative translation along z axis can also be approached from homography parameters. This has been previously experimented with a Labmate robot [14]. It turns out more difficult to obtain lateral translation along x axis. Using other approaches [6] or using the complete algorithm (§2) the translation is obtained up to a scale factor and the correction should be made in two steps, for example using the displacement provided by odometry. However, the heading correction with μ compensates directly lateral translation, as it will be explained below.

4 Performance of motion from homography

We have made many experiments with synthetic and real data, and we have compared the heading correction computed from the complete motion algorithm and those approached from μ , t or eigenvalues.

A simulator of scenes and motions has been built. The image noise and other not considered errors have been simulated as gaussian noise added to lines coordinates. The influence of random noise, zero mean and one pixel standard deviation in a typical situation can be appreciated in figure 2. Parameter μ is much less affected by noise (Fig. 2.a) and therefore the estimation of rotation from μ will be better requiring only the calibration of a parameter. We can see how t is highly affected by noise (Fig. 2.b). The behavior of the rotation estimate from eigenvalues is also affected by noise, but it has the advantage of not being dependent of internal camera parameters (Fig. 2.c). Finally it can be appreciated that noise barely affects rotation computed with the complete algorithm (Fig. 2.d).

We have also made experiments to show how combined motions could affect rotation error depending on the way to obtain it (Fig. 3). Translation along x axis affects t the least but t is the most noisy. On the other side, it affects μ the most but μ is the least noisy (Fig. 3.a).

The effect of translation on μ , which could be interpreted as a drawback, it is in practice an advantage. This is because total translation (without scale factor) is difficultly computed without additional information. However, the over-rotation obtained with μ allows to correct the displacement generated in the x direction when vehicle advances without its explicit computation, supposing the images are acquired approximately in the direction of advance. This correction will be higher with small scene depth.

With respect to translation along z axis, it has been confirmed that it barely affects rotation error for all the methods, μ being the most stable (Fig. 3.b).

A long term navigation has been also simulated. In figure 4 we can see the trajectory of the robot along a corridor making a heading correction every seven meters approximately. We compare μ and the complete motion algorithm. When there is only error in rotation, both ways to obtain the heading correction behave similarly (Fig. 4.a). When there is also lateral deviation, we can see how the heading correction obtained with the complete motion algorithm barely compensate the lateral deviation (Fig. 4.b). However, when heading is corrected with μ , the robot behaves better because the

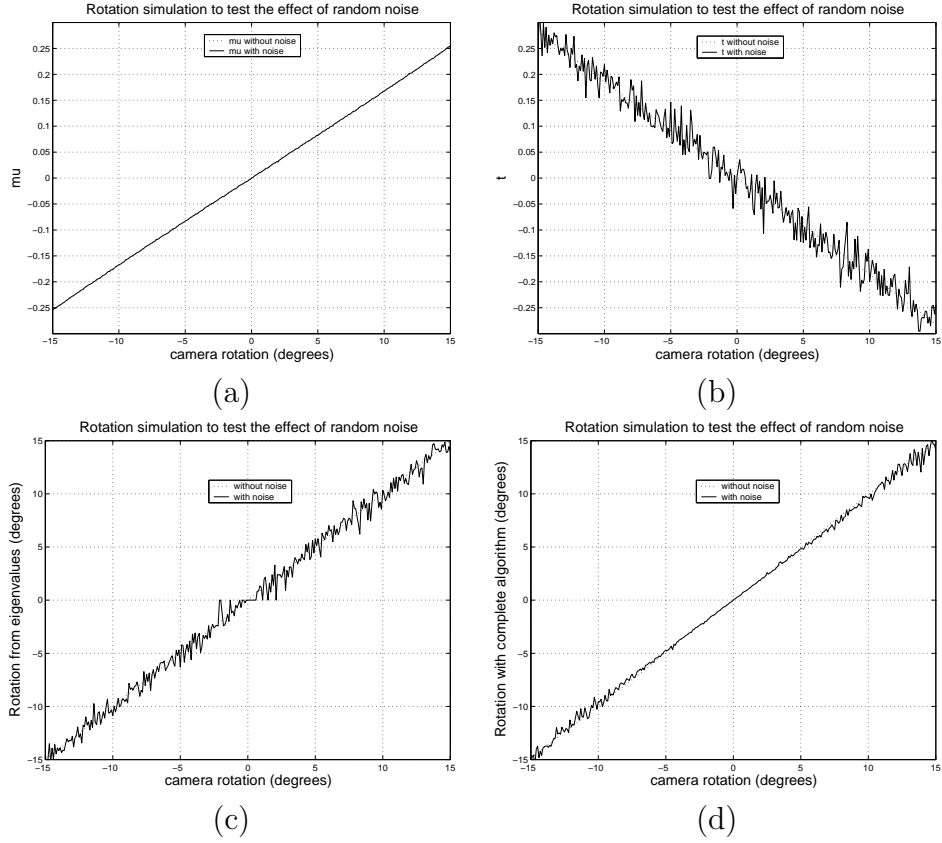


Figure 2: Effect of gaussian random noise (zero mean and one pixel standard deviation) on the parameters of the homography used to compute rotation. (a) Effect on μ , (b) effect on t , (c) effect on rotation from eigenvalues θ , and (d) effect on rotation from complete motion algorithm.

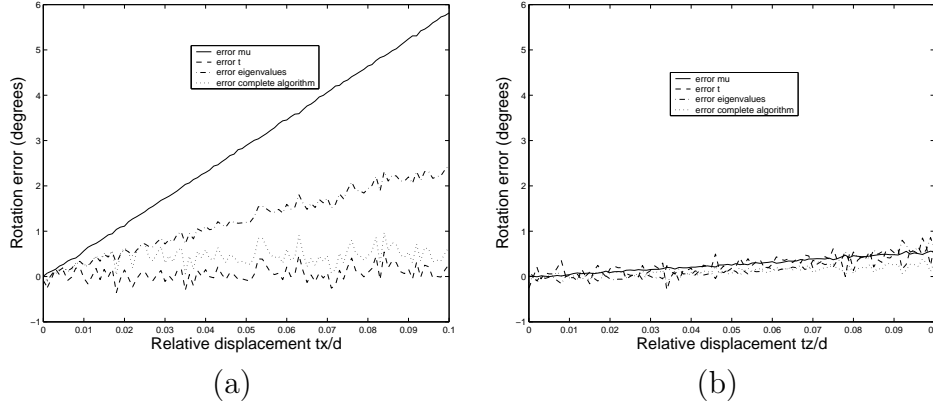


Figure 3: Effect of relative translation on the rotation estimation error for the four possible ways to compute it. (a) Relative translation along x axis, (b) relative translation along z axis. Gaussian noise in the image of 0.5 pixels standard deviation has been considered and the camera rotates 5 degrees.

Average error	μ	t	e.v.	c. a.
Planar scene (Fig. 5.a-b)	0.056	0.372	0.909	0.130
Non planar scene (Fig. 5.c-d)	0.068	0.880	0.997	0.075

Table 1: Average error in degrees for the rotation computed with μ , t , eigenvalues (e.v.) and complete motion algorithm (c.a.).

over-correction of rotation compensates the lateral translation (Fig. 4.b). This correction is more evident at the end of the corridor when the scene depth is smaller.

4.1 Motion using real images

We have made performance experiments with real images (Fig. 5) computing automatically motion from them. The camera has been moved with a precision head (UTR80, Manufactured by Newport, Resolution $1/60^\circ$) which provides the ground truth. Different cases have been considered and we show results of two of them: Planar scene which is approximately rotated 25 degrees respect to image plane (Fig. 5.a) and scene composed by two perpendicular planes (Fig. 5.b).

The camera used in these experiments has a focal length of 6 *mm.*, a CCD of $1/2''$ and an image size of 576×768 pixels. In all the cases camera

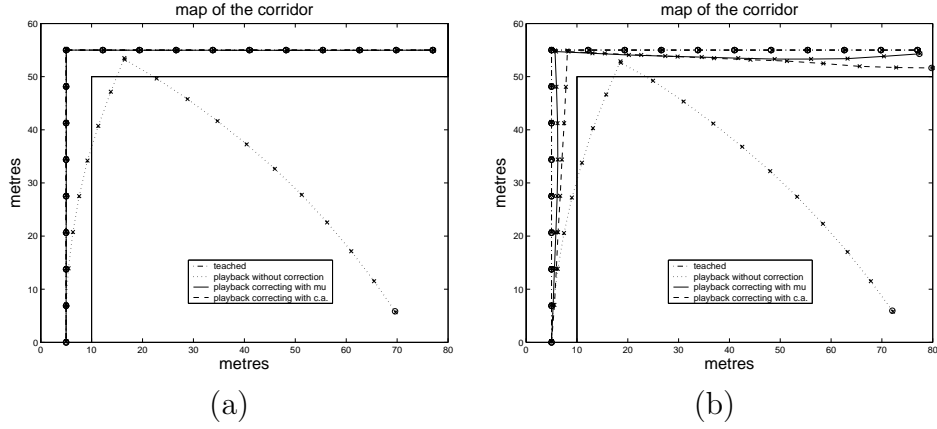


Figure 4: The robot moving in a corridor adding to motion two types of gaussian error. (a) Error in rotation (0.05 radians mean and 0.01 standard deviation every step). (b) Error in rotation (0.05 radians mean and 0.01 standard deviation) and error in lateral translation (mean 5% of robot advance and standard deviation of 1%). In both cases we show the trajectories when there is no correction, when heading is corrected from μ and when heading is corrected from the complete motion algorithm.

rotates from -10 to 10 degrees, extracting and matching automatically lines and computing homographies respect to the central image (a total of 29 images of each scene has been used). In table 1 the average error is shown. With μ parameter and with the complete motion algorithm, it turns out an error similar to the resolution of the head used to move the camera. As the angle is relatively short, $\tan(\theta) \approx \theta$, being then possible to linearly fit μ and t in relation to camera rotation to obtain the calibration parameter. In figure 6 we show the computed μ parameter in one of the scenes previously commented, being similar in the other.

It has been proved also that the vision matching is robust providing good and stable homography computation. With respect to the different ways to compute heading from homography, it is confirmed the results advanced by simulations, that we summarize as follows:

- Random noise and inaccuracies in extraction of features in real images affects t and eigenvalues the most, lesser to rotation computed with the complete motion algorithm and it nearly does not affect μ .
- Translations along x axis affect μ the most. This confirms that transla-



(a)



(b)



(c)



(d)

Figure 5: Examples of the images (with automatically matched lines) used for the accuracy evaluation.

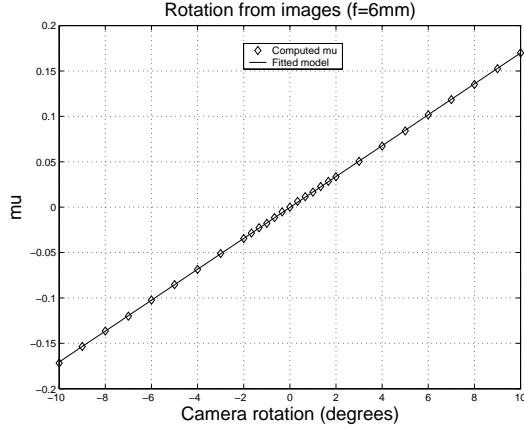


Figure 6: Computed μ from real images in the case of planar scene and its linear fitting. In the other cases the behavior is similar.

tion deviation along x axis is coupled with rotation, being this coupling captured by μ . In the playback phase this turns out to be good, because heading correction through parameter μ produces an over-rotation that compensates the translation deviation, supposing the images are acquired in the direction of advance.

- Errors in translation along the direction of advance (z axis) barely affect μ , being therefore a good parameter to estimate robot heading.

Let us remember that the rotation computed with eigenvalues or complete motion algorithm gives two solutions, being a problem in practice to select the correct one. In all the experiments of the laboratory presented above we have selected the solution closest to the correct one.

5 Integration into an autonomous wheelchair

Finally to test the visual system in a real environment we have used an electrical wheelchair. In order to correct the heading while the wheelchair is moving some adaptations of the image processing have been made. Besides, the correction has been integrated into the navigation system based on potential fields, allowing to carry out smooth trajectories. This is specially useful because the wheelchair has problems to make pure rotations due to its castor wheels.

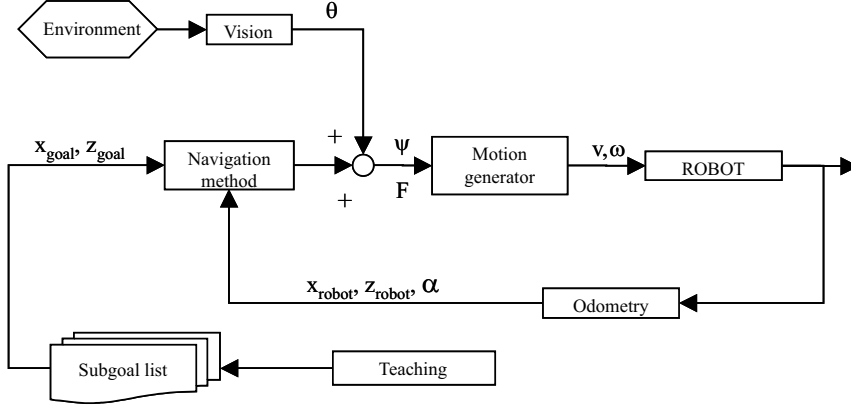


Figure 7: Global scheme of the wheelchair navigation system.

5.1 Navigation system

Artificial potential fields are probably the most widely studied methods for mobile robot navigation. The advantages of potential fields are the conceptual simplicity, the medium-high reliability in all kind of robots, the easy implementation and the smoothness of the final trajectories. We use the navigation system presented in [15], which has been designed for robots with differential drive, allowing the obstacle avoidance in real-time. It considers a simple physical model of the robot for the computation of the trajectory. The state space model can be written as,

$$\dot{\mathbf{x}} = -2b \begin{bmatrix} 1 & 0 \\ 0 & k_i \end{bmatrix} \mathbf{x} + F \begin{bmatrix} 1 & 0 \\ 0 & k_i h \end{bmatrix} \begin{bmatrix} \cos \psi \\ \sin \psi \end{bmatrix} \quad (3)$$

where $\mathbf{x} = (v, w)^T$ is the velocity state (linear and angular), b is the viscous friction coefficient, h is the distance between the mass center and the force application point and $k_i = m/I_{zz}$ is the ratio between the mass and the inertia moment around the vertical axis. Additionally, F is the attractive force of the goal whose direction is ψ .

As it can be seen in the scheme of the global navigation system (Fig. 7), we correct the direction of the attractive force with θ , instead of correcting robot orientation α . So, the maneuvers to compensate localization errors are automatically and gradually applied to the trajectory computed by the navigation method. Moreover, as we have exposed previously, this maneuver include an over-rotation, which in a non-holonomic robot corrects lateral

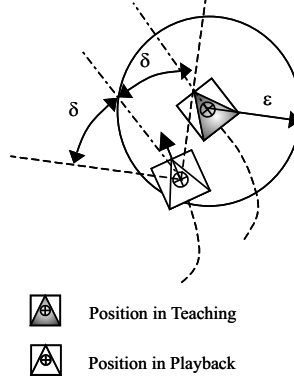


Figure 8: Geometric test to match the images obtained in the teaching and in the playback stages.

displacement.

To determine that the position reached by the robot is close enough to the point where the original image was acquired, two geometric test are carried out (Fig. 8). The robot should reach to the goal position (distance less than ε) with similar orientation (less than $\pm\delta$). In our tests the minimum requirements for the system to work were $\varepsilon = 0.7$ m. and $\delta = 20$ degrees.

5.2 Navigation experiments

We are using a wheelchair with a camera, whose image plane is approximately vertical, placed on its front. Several experiments have been carried out, and the first presented here consist on a trip of several turns around a room. For the teaching phase, four images have been taken (Fig. 9). The trajectory (Fig. 10) is labelled with capital letters to show the positions during the teaching phase and with lower letter to show the positions reached by the wheelchair when motion is achieved only with odometry. In this case the number indicates the number of turn. After 5 laps the error is bigger than 4 meters in position and 2.9 radians in orientation. However when the heading correction presented in this work is introduced in the navigation system, total error is negligible. Figure 11 shows the position of the wheelchair at different rounds of the room when it comes back to the initial location.

The data referring number of lines, matches and computing time are shown in table 2. As the number of lines extracted is reasonably high, it turns out that the process also works when some change appears in the scene such



Figure 9: Reference images for the teaching phase with the extracted lines superimposed (after lens distortion correction). Each image is taken at a corner and in the direction of the next reference location.

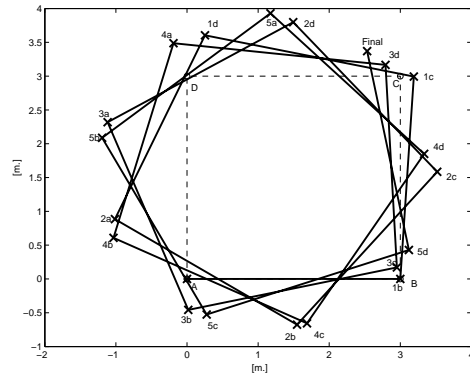


Figure 10: Positions on the teaching phase (capital letters) and positions reached without correction of motion (lower letters) in several turns (numbers).

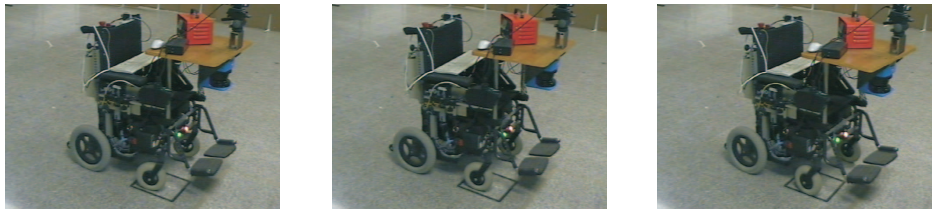


Figure 11: Position of the wheelchair after three, four and five rounds to the room. It can be observed that the differences are small enough.

Image Size	288x768
Number of lines image 1	51
Number of lines image 2	50
Possible matches	26
Computed robust matches	17
Extraction of lines	111 ms.
Matching	32 ms.
Computation of homography	18 ms.
Total time	161 ms.

Table 2: Some details on number of lines, matches and processing time in milliseconds (ms.).



Figure 12: Reference image and the corresponding image captured at third turn of the experiment. Some changes have appeared in the scene in the meantime, but visual robustness remains.

as illumination changes or people moving around (Fig. 12). The processing time of the visual correction is about 150 milliseconds using non-optimized ANSI C on a PIII, 850Mhz, 512Mb RAM, which is enough to make corrections of motion as wheelchair moves. Anyway, considering the wheelchair dynamics, our correction system turns out to be stable for sampling rates about one second.

To solve the displacement between frames in images captured in motion, we process only even rows of the image. On the other hand, to obtain more working range, a grand angular lens ($f = 3mm.$) has been used. Correction of radial distortion, however, is needed to deal with this lens. A previous calibration was used to correct the coordinates of the lines extracted in the original images. It has been considered the classical single parameter radial distortion model. An example of the influence of lens distortion can be seen when the original image is compared with the location of the lines after distortion correction (Fig. 13).



Figure 13: Acquired image and superposed line features after radial distortion correction.

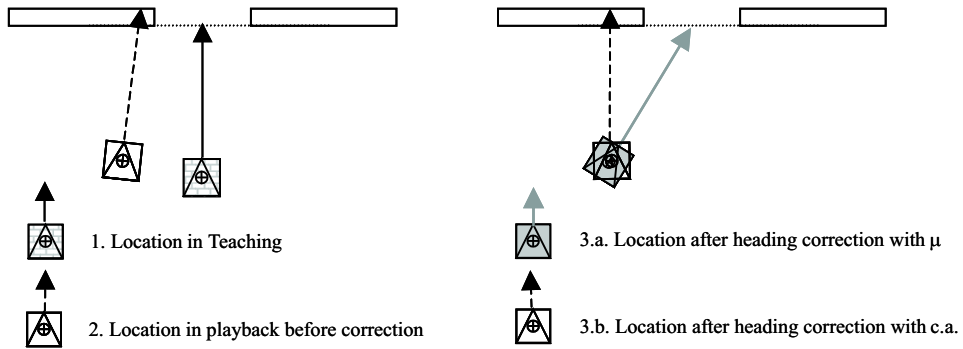


Figure 14: The effect of scene depth becomes evident when crossing a door. The heading correction provided by μ makes the robot to point to the center of the door even there is lateral displacement. When the heading is corrected with the complete motion algorithm, translation estimation will be needed to avoid collision because there is not enough over-rotation to compensate it.

Crossing doors is a hard test for most mobile robots. A little error during the approach maneuver normally means a crash with the door jambs. However with our approach the robot will cross the door just by the middle using a reference image taken in a suitable location (Fig. 13). The explanation of this phenomenon is a special case of the over-rotation effect with parameter μ , which happens when the assumption of far scene (at infinitum) is not true. In fact, the homography tries to match both scenes, which means that the correction is the rotation needed to align the center of the mobile camera (the robot) with the center of the reference image, that is the door (Fig. 14).

From <http://webdiis.unizar.es/~jguerrer/mapless/mapless.htm>, a demo of the experiments and a program to match lines computing heading simultaneously can be downloaded.

6 Conclusions

We have presented a map-less navigation approach, that uses previously taken images to correct robot motion using lines as key features. The lines are automatically extracted and matched in a robust way, computing simultaneously an homography which is the basis to correct robot heading.

Although total motion, rotation and translation with a scale factor could be computed, we propose to use the parameters of the homography to correct heading. In particular the μ parameter is robust to noise and it captures the effect of lateral translation, in such a way that it can be compensated when robot advances without two steps correction, nor additional sensors.

The proposal has been tested with synthetic data and real images seeing its robustness. Additionally, a real time implementation with a wheelchair for handicapped has been made with success. It has been seen that, whenever a previous teaching phase were made to take the reference images, the wheelchair may remain in a playback phase correcting heading, and compensating simultaneously lateral translations.

References

- [1] G.N. DeSouza and A. C. Kak. Vision for mobile robot navigation: A survey. *IEEE Trans. on Patt. Analysis and Machine Intelligence*, 24(2):237–267, 2002.

- [2] J. Santos-Victor, G. Sandini, F. Curotto, and S. Garibaldi. Divergent stereo in autonomous navigation: From bees to robots. *Int. Journal on Computer Vision*, 14(2):159–177, 1995.
- [3] Y. Matsumoto, M. Inaba, and H. Inoue. Visual navigation using view-sequenced route representation. In *IEEE Int. Conf. Rob. and Autom.*, pages 83–88, 1996.
- [4] J.J. Guerrero and C. Sagüés. Uncalibrated vision based on lines for robot navigation. *Mechatronics*, 11(6):759–777, 2001.
- [5] A. Ohya, Y. Miyazaki, and S. Yuta. Autonomous navigation of mobile robot based on teaching and playback using trinocular vision. In *Proceedings of the 27th IEEE Industrial Eletronics Conference (IECON'01)*, pages 398–403, 2001.
- [6] R. Basri, E. Rivlin, and I. Shimshoni. Visual homing: Surfing on the epipoles. In *IEEE Conference on Computer Vision*, pages 863–869, 1998.
- [7] B. Liang and N. Pears. Visual navigation using planar homographies. In *IEEE Conference on Robotics and Automation*, pages 205–210, 2002.
- [8] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge, 2000.
- [9] D. Ortin and J. M. M. Montiel. Indoor robot motion based on monocular images. *Robotica*, 19:331–342, 2001.
- [10] J.J. Guerrero and C. Sagüés. Robust line matching and estimate of homographies simultaneously. In *IbPRIA, Pattern Recognition and Image Analysis, LNCS 2652*, pages 297–307, 2003.
- [11] Z. Zhang. Parameter estimation techniques: A tutorial with application to conic fitting. Rapport de recherche RR-2676, I.N.R.I.A., Sophia-Antipolis, France, 1995.
- [12] P.J. Rousseeuw and A.M. Leroy. *Robust Regression and Outlier Detection*. John Wiley, New York, 1987.
- [13] J. Weng, T.S. Huang, and N. Ahuja. *Motion and Structure from Image Sequences*. Springer-Verlag, Berlin-Heidelberg, 1993.

- [14] C. Sagüés and J.J. Guerrero. Visual correction for mobile robot homing. *Robotics and Autonomous Systems*, 50:41–49, 2005.
- [15] J. R. Asensio and L. Montano. A Kinematic and Dynamic Model-Based Motion Controller for Mobile Robots. In *15th World Congress of the IFAC*, Barcelona, Spain, July 21th-26th 2002.