

Position-Based Navigation Using Multiple Homographies

Eduardo Montijano *
DIIS - I3A
University of Zaragoza, Spain
emonti@unizar.es

Carlos Sagues
DIIS - I3A
University of Zaragoza, Spain
csagues@unizar.es

Abstract

In this paper we address the problem of visual navigation of a mobile robot which simultaneously obtains metric localization and scene reconstruction using homographies. Initially, the robot is guided by a human and some scenes during the trip are stored from known reference locations. The interest of this paper consist in the possibility of getting real and precise data of the robot motion and the scene, which presents some advantages over other existing approaches. For example, it allows the robot to carry out other trajectories than the executed during the teaching phase. We show an extensive analysis of the output in presence of errors in some of the inputs.

Index Terms - homography, visual navigation, motion estimation, reactive navigation, metric reconstruction.

1. Introduction

It is well known that the idea of autonomous robot navigation is of great interest for humans. The use of visual information for this task is chosen for two reasons. First, the use of visual sensor provides big amount of information that is contained in the scene and second, its low cost compared with other sensors.

Reviewing the literature, multiple solutions addressing this problem can be found. One of the earliest is presented in [9], where a sequence of images with a description of the motion is given to the robot and a direct matching between the pixels of the images is used. The recently appearance of invariant features in the images [7] [1] has given more robustness and better matching, allowing larger displacements between the compared images. The authors of [4] propose a qualitative navigation method with good results. In [2] a topological map formed by reference images is used to perform different trajectories using the homography matrix. These solutions have the problem that the trajectories performed by the robot must follow the sequence of the reference images. If for any reason some obstacles appear, and the robot gets lost from

the prefixed path, it will not be able to recover it and it will not arrive to the goal.

In this contest of image based approaches, the proposal presented in [12] provides good information about orientation and lateral deviation errors. Solutions like the ones found in [3], [8] and [11] develop specific control laws to move the robot to the goal, specified by a target image, with high precision. The main advantage of these approaches is that they do not need neither information about the scene nor the positions of the images. However, its application will be limited to the trajectories that move the robot from any position to the target position using the same route than the used in the teaching phase.

We propose here a technique that obtains metric reconstruction using the homography matrix. The metric reconstruction needs the depth of the planes in the scene and their orientation. The main disadvantage is that several parameters of the scene must be given. The algorithms proposed here only need one of these two parameters and easily computes the other taking advantage of the multiple images stored during the teaching phase where the robot is moved along a path. We present an extensive analysis of the influence of the input errors, which is confirmed with several simulations.

The outline of this paper is as follows. After showing the Homography Sensor in section II, the methods to improve observations when there is more than one homography are presented in section III. Section IV details the errors induced by wrong inputs. In section V some simulations show the results of this technique. Finally, some concluding remarks and future lines of work are exposed in section VI.

2 The Homography Sensor

First of all, let us show the notation used in this paper. If we consider one parameter a , we denote its estimated value as \hat{a} and an estimation error as \tilde{a} . Matrix are denoted with capital bold letters (**M**) and vectors with bold lower-case (**u**).

Let us consider a non holonomic unicycle robot moving on the ground plane XZ. Due to the planar motion only three parameters are needed to express the robot state, $\mathbf{s} = (x, z, \theta)^T$. If we refer only to the position of the

*This work was supported by projects DPI 2006-07928, IST-1-045062-URUS-STP and grant B116/08 Gobierno de Aragón

robot we will denote it as $\mathbf{p} = (x, z)^T$. The considered robot has fixed onboard a monocular camera. Since the robot is moving indoors, most part of the information acquired will correspond to views of walls and planes of the environment. We assume that all the visible planes are perpendicular to the plane where the robot is moving, which is common inside buildings. With this assumption we can represent a generic plane π only with two parameters, its distance to the origin and its orientation, $\pi = (d_\pi, \alpha)$. The normal vector of the plane then is $\mathbf{n} = (-\cos \alpha, 0, \sin \alpha)^T$, see Fig. 1.

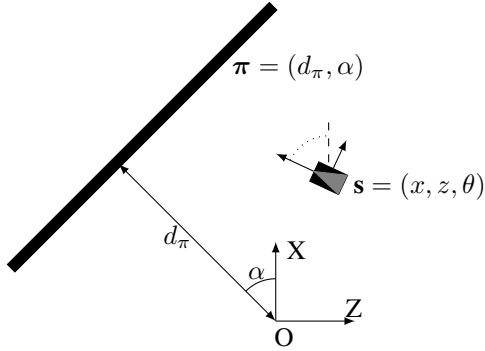


Figure 1. Coordinate systems and plane definition

Consider the case where the robot state \mathbf{s}_2 is unknown and we have an image of a plane captured from a known reference position \mathbf{s}_1 . If the image taken from \mathbf{s}_2 has at least four corresponding matches with the image acquired at \mathbf{s}_1 , it has been proved that there is a relation between these images expressed by a 3x3 homography matrix \mathbf{H}_{21} [6]. The Homography Sensor (HS in the following), is a function $HS : \mathbb{R}^{11} \rightarrow \mathbb{R}^3$ that allows the robot to estimate its position $\mathbf{s}_2 = HS(\mathbf{s}_1, \mathbf{H}_{21}, \mathbf{K}, \pi)$ using as inputs the reference position \mathbf{s}_1 , the homography matrix \mathbf{H}_{21} , the intrinsic calibration parameters of the camera \mathbf{K} and the plane information π . The homography \mathbf{H}_{21} is represented by the four parameters $\{h_{11}, h_{13}, h_{31}, h_{33}\}$, the other five parameters of the matrix are not needed in the function; the calibration matrix \mathbf{K} is determined by f_x , the focal length in x direction measured in pixels and x_0 the x-coordinate of the principal point in the image.

HS can be computed in two steps. The first one calculates \mathbf{s}_{12} , the coordinates of \mathbf{s}_1 expressed in \mathbf{s}_2 reference, by

$$\mathbf{s}_{12} = \mathbf{s}_{12}(\mathbf{H}, \mathbf{K}, \pi) = \begin{pmatrix} d_\pi(n_x \psi_1 + n_z \psi_2) \\ d_\pi(-n_z \psi_1 + n_x \psi_2) \\ \arctan(\psi_3, \psi_4) \end{pmatrix} \quad (1)$$

where ψ_i are given by the following expressions

$$\begin{cases} \psi_1 = 2x_0 h_{31} + h_{33} - h_{11}, \\ \psi_2 = -f_x^{-1} \psi_5 - f_x h_{31}, \\ \psi_3 = -n_x n_z \psi_1 - n_x^2 \psi_5 + n_z^2 \alpha_x h_{31}, \\ \psi_4 = -n_x n_z \psi_2 - n_x^2 (h_{31} + h_{33}) + n_z^2 (h_{11} - x_0 h_{31}), \\ \psi_5 = x_0 h_{11} - x_0^2 h_{31} + h_{13} - x_0 h_{33}, \end{cases} \quad (2)$$

$n_x = \cos \alpha$ and $n_z = \sin \alpha$. The second step calculates \mathbf{s}_2 , obtained from composition of \mathbf{s}_1 and \mathbf{s}_{12} .

If we make a more extensive analysis of (1) it is possible to see two details. First one is that the orientation estimation is independent of the distance to the plane. This means that we will be able to obtain precise values of θ without the necessity of knowing the distances to the reference plane. The second detail is that the pose estimation, $\mathbf{p}_{12} = (x_{12}, z_{12})^T$, can be rewritten to the form

$$\mathbf{p}_{12} = d_\pi \mathbf{R}_\alpha \psi, \quad (3)$$

with $\psi = (\psi_1, \psi_2)^T$ (2) and \mathbf{R}_α a 2x2 rotation matrix of α units. This means that the pose estimation is calculated from a vector ψ , scaled and rotated depending on the plane π . This decoupled representation will help us to improve the pose estimations.

The main problem of the HS is the amount of needed input parameters. The goodness of the method depends on the goodness of the values of these input parameters. Here we will present methods to avoid errors taking advantage of the views of multiple planes in the same image.

2.1 Camera calibration

In this paper we will not deal with the process of obtaining the intrinsic parameters of the camera and assume them to be well obtained by a previous offline step [6] [5] [10]. In order to do that we refer to the recent self-calibration method proposed by Menudet et al. [10]. We have chosen this method for its relation with our work with homographies besides its simplicity to be implemented.

This method tries to minimize the eigenvalues of the 2x2 top-left block of

$$\mathbf{S}_{k1}^T \mathbf{S}_{k1} = \mathbf{N}_1^T \bar{\mathbf{H}}_{k1}^T \bar{\mathbf{H}}_{k1} \mathbf{N}_1, \quad (4)$$

being $\bar{\mathbf{H}}_{k1}$ the normalized homography between the image k and the reference image, and \mathbf{N}_1 the rotation matrix that transforms the vector $\mathbf{n} = (0, 0, 1)^T$ in the vector that represents the normal to the plane in the position where the reference image was acquired. Note that in our case this matrix has only one degree of freedom because we assume that the reference plane and the plane of the captured image are perpendicular. \mathbf{S}_{k1} is a 3x3 matrix that represents a planar similarity with 4 degrees of freedom. It relates the unprojected points of the images in a virtual plane parallel to the real one. Additionally, this method also allows to estimate the normals to the plane in the reference positions.

Since the calibration parameters will still have some errors, in the simulations we will show an example with errors in \mathbf{K} matrix and we will show that these errors do not affect much to the estimated position. We also dismissed the possible errors obtained in the calculation of the homography since they will be too small in comparison with the possible errors introduced in the estimations of the remaining parameters.

3 Multiple plane observations

Since the robot will be moving indoors all the time, many of the images acquired will contain information of more than one plane. It is possible to take advantage of this situation computing multiple homographies. Considering this fact we combine multiple observations in order to improve the estimations. The methods proposed here are based on geometric rules, are robust to possible errors in the inputs and also are able to decrease these possible errors so that future estimations will not be affected by them. We propose two methods, the first is based on intersection of lines and the second one is based on intersection of circumferences.

Let us suppose that in a previous teaching phase the robot has moved following a path in the environment and has acquired a list of images at known positions $\mathbf{s}_i \in \mathbb{N}$. During the navigation we assume that the robot is at an unknown position where the it is possible to compute two homographies corresponding to two of the reference images. For an easy understanding, in the following we denote the unknown position as \mathbf{s}_3 , the reference positions as \mathbf{s}_1 and \mathbf{s}_2 and the corresponding homographies \mathbf{H}_{3i} and \mathbf{H}_{3j} . The advantage of having two homographies is that it is possible to relax the required information about the planes. Instead of having two parameters to describe each plane, only one of them, distance or orientation, is required. Depending on the missing parameter one method or the other will be used.

3.1 Depth estimation method

Depth estimation method (*DEM*) can be defined as a function

$$\begin{aligned} DEM : \mathbb{R}^{18} &\rightarrow \mathbb{R}^5, \\ (\mathbf{s}_1, \mathbf{s}_2, \mathbf{H}_{31}, \mathbf{H}_{32}, \mathbf{K}, \alpha_1, \alpha_2) &\rightarrow (\mathbf{s}_3, d_{\pi 1}, d_{\pi 2}), \end{aligned} \quad (5)$$

being $(d_{\pi 1}, \alpha_1)$ the distance and orientation of one reference plane with respect to \mathbf{s}_1 and $(d_{\pi 2}, \alpha_2)$ the distance and orientation of other reference plane with respect to \mathbf{s}_2 . The reference plane can be the same for the two images or two different planes, one for each image. As it is seen in (5), besides the computation of the actual position of the robot \mathbf{s}_3 the distances of the reference positions to the planes are provided. Let us note that whereas in the function *HS* the distance was required as an input, here it is obtained as an output parameter.

The main idea is the following. As we have seen in eq. (3), depth acts as a scale factor in the estimation of

positions \mathbf{p}_{31} and \mathbf{p}_{32} . Since $d_{\pi 1} > 0$ and $d_{\pi 2} > 0$, we can give them an initial value, for example $\hat{d}_{\pi 1} = \hat{d}_{\pi 2} = 1$ and use this values in (1) to compute \mathbf{s}_{31} and \mathbf{s}_{32} . With this information we know that \mathbf{s}_3 will lay in the straight line that cross \mathbf{s}_1 and \mathbf{s}_{31} , which can be expressed as

$$r_1 \equiv \mathbf{s}_1 + \lambda_1(\mathbf{s}_{31} - \mathbf{s}_1). \quad (6)$$

We also know that \mathbf{s}_3 belongs to other line defined in a similar way but using the parameters of the second reference position,

$$r_2 \equiv \mathbf{s}_2 + \lambda_2(\mathbf{s}_{32} - \mathbf{s}_2), \quad (7)$$

so the position \mathbf{s}_3 will correspond to the intersection of r_1 and r_2 ,

$$\mathbf{s}_3 = \begin{bmatrix} A_1 & B_1 \\ A_2 & B_2 \end{bmatrix}^{-1} \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} \quad (8)$$

with

$$\begin{cases} A_1 = s_{31x} - s_{1x} & A_2 = s_{32x} - s_{2x} \\ B_1 = s_{1z} - s_{31z} & B_2 = s_{2z} - s_{32z} \\ C_1 = -(A_1 s_{1x} + B_1 s_{1z}) & C_2 = -(A_2 s_{2x} + B_2 s_{2z}). \end{cases} \quad (9)$$

As we can observe in the equation, this technique only works when \mathbf{s}_1 , \mathbf{s}_2 and \mathbf{s}_3 are not collinear ($r_1 \neq r_2$). The other singularity case would be if r_1 and r_2 were parallel but this is not possible since they always share a common point \mathbf{s}_3 .

Once the true \mathbf{s}_3 is known, we are also able to compute the actual distances of \mathbf{s}_1 and \mathbf{s}_2 to the planes with the following equations,

$$\begin{aligned} d_{\pi 1} &= \hat{d}_{\pi 1} |\mathbf{s}_3 - \mathbf{s}_1| / |\mathbf{s}_{31}| \\ d_{\pi 2} &= \hat{d}_{\pi 2} |\mathbf{s}_3 - \mathbf{s}_2| / |\mathbf{s}_{32}| \end{aligned} \quad (10)$$

The resulting method is simple and it does not deal with any of the camera orientations where the images are taken. The estimation of the robot orientation in \mathbf{s}_3 will not be affected by the errors of the method. It is a good technique to combine with the self-calibration method chosen [10] because it computes the remaining input parameters.

3.2 Normal-Orientation estimation method

In other cases, the distances to the planes can be better known than the normals. In these situations the Depth Estimation Method will not work properly. The solution we propose here has the form

$$\begin{aligned} NOEM : \mathbb{R}^{18} &\rightarrow \mathbb{R}^5, \\ (\mathbf{s}_1, \mathbf{s}_2, \mathbf{H}_{31}, \mathbf{H}_{32}, \mathbf{K}, d_{\pi 1}, d_{\pi 2}) &\rightarrow (\mathbf{s}_3, \alpha_1, \alpha_2). \end{aligned} \quad (11)$$

As in the previous method an initial estimation of the unknown angles of the plans is required. In this case, for simplicity we can take $\hat{\alpha}_1 = \hat{\alpha}_2 = 0$, so $\mathbf{R}_{\alpha i} = \mathbf{I}$ and $\hat{\pi}_i \perp \mathbf{s}_i$, and compute \mathbf{s}_{31} and \mathbf{s}_{32} with eq. (1).

Looking again at (3) we see that uncertainty of the normal to the plane implies that \mathbf{s}_3 can be any point situated

on the circumference of center the reference point and radius $|\mathbf{s}_{3i} - \mathbf{s}_i| = |d_{\pi i} \boldsymbol{\psi}_i|$, $i = \{1, 2\}$. We define then the circumferences:

$$\begin{aligned} C_1(\mathbf{s}_1, |d_{\pi 1} \boldsymbol{\psi}_1|) &\equiv (X - s_{1x})^2 + (Z - s_{1z})^2 = |d_{\pi 1} \boldsymbol{\psi}_1|^2 \\ C_2(\mathbf{s}_2, |d_{\pi 2} \boldsymbol{\psi}_2|) &\equiv (X - s_{2x})^2 + (Z - s_{2z})^2 = |d_{\pi 2} \boldsymbol{\psi}_2|^2. \end{aligned} \quad (12)$$

The value of \mathbf{s}_3 will be one of the intersection points of C_1 and C_2 . Let us note that in this case the orientation of the robot is still wrong unless we compute the actual normals to the planes. In order to obtain the values of the normals, we propose an iterative method using the previous information. If the robot orientation was computed correctly even with bad inputs of the normals, the right values of the angles α_1 and α_2 would be

$$\begin{cases} \alpha_1 = \hat{\alpha}_1 + \angle([\mathbf{s}_1, \mathbf{s}_{31}], [\mathbf{s}_1, \mathbf{s}_3]) \\ \alpha_2 = \hat{\alpha}_2 + \angle([\mathbf{s}_2, \mathbf{s}_{32}], [\mathbf{s}_2, \mathbf{s}_3]), \end{cases} \quad (13)$$

where $[\mathbf{s}_i, \mathbf{s}_j]$ denotes the line formed by the points \mathbf{s}_i and \mathbf{s}_j . Since the points \mathbf{s}_{31} and \mathbf{s}_{32} are expressed in reference coordinates they depend on the estimated orientation, but the computed orientations, θ_{31} and θ_{32} , are wrong, so α_1 and α_2 are wrong too. However the new values of α_1 and α_2 will be closer to the real ones, so it is possible to compute new orientations with the *HS* and repeat the process until the difference between the α_i obtained iteratively are close enough.

As it is known, there are two possible solutions to the problem of the intersection of two circumferences. Once both possibilities are computed we have to discern what of both solutions is the true one. We do this by comparing the two new orientations of the robot obtained applying *HS*. The good solution will be the one that gives the same orientation for the robot using the α_i computed angles.

The main advantages of this method are that it will work although the three points are collinear, in this case the intersection of C_1 and C_2 will return only the true solution. It will not work if the two centers correspond to the same point (which is impossible with our hypothesis) or if the two radius are too short so that the circumferences do not intersect, which will mean that \mathbf{K} , $d_{\pi i}$ or both are badly estimated. The main problem of this method is that it needs to compute the actual normals and the orientation of the robot using an iterative algorithm. In our simulations we have seen that no more than five iterations are required to compute the final value of the angles. Other problem we have found is that it does not work properly when one of the reference positions is too close to the unknown position.

4 Errors in the inputs

The methods presented here assume the goodness of the input parameters. In this section we have analyzed how some errors in the inputs affect to the values of the outputs using *DEM* equations. Specifically we have focused on errors in the reference positions and in the normals to the planes, which probably will be the parameters

with greater errors. Similar analysis can be done to the *NOEM* equations.

4.1 Errors in the reference positions

We show here how the *DEM* will be affected by errors in the reference position. We define the errors in these inputs as

$$\tilde{\mathbf{s}}_1 = \mathbf{s}_1 - \hat{\mathbf{s}}_1, \quad \tilde{\mathbf{s}}_2 = \mathbf{s}_2 - \hat{\mathbf{s}}_2. \quad (14)$$

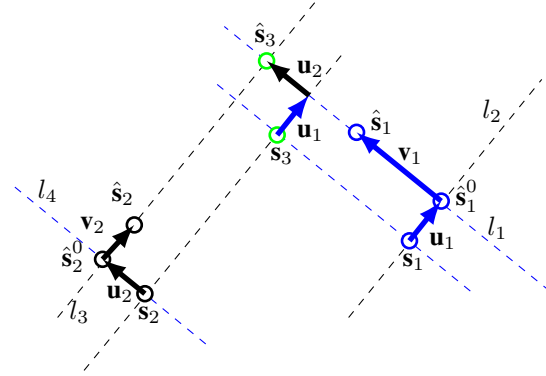


Figure 2. Error obtained in \mathbf{s}_3 estimation when there are errors in the reference systems. It is possible to observe that $\tilde{\mathbf{s}}_3 = \mathbf{u}_1 + \mathbf{u}_2$

We show graphically the induced errors. Looking at Fig. 2 we observe that the error in \mathbf{s}_3 estimation can be expressed as the sum of two vectors \mathbf{u}_1 and \mathbf{u}_2 . In order to obtain the analytical expressions of these vectors we define $\hat{\mathbf{s}}_1^0$ as the point where the lines $l_1 \equiv \hat{\mathbf{s}}_1 + \lambda(\hat{\mathbf{s}}_3 - \hat{\mathbf{s}}_1)$ and $l_2 \equiv \mathbf{s}_1 + \lambda(\hat{\mathbf{s}}_3 - \hat{\mathbf{s}}_2)$ intersect. We also define $\hat{\mathbf{s}}_2^0$ as the intersection of $l_3 \equiv \hat{\mathbf{s}}_2 + \lambda(\hat{\mathbf{s}}_3 - \hat{\mathbf{s}}_2)$ and $l_4 \equiv \mathbf{s}_2 + \lambda(\hat{\mathbf{s}}_3 - \hat{\mathbf{s}}_1)$. Once these two points have been computed the desired vectors will be

$$\mathbf{u}_1 = \hat{\mathbf{s}}_1^0 - \mathbf{s}_1, \quad \mathbf{u}_2 = \hat{\mathbf{s}}_2^0 - \mathbf{s}_2, \quad (15)$$

and the expression of $\tilde{\mathbf{s}}_3$ is given by

$$\tilde{\mathbf{s}}_3 = \mathbf{u}_1 + \mathbf{u}_2. \quad (16)$$

It is clear that $|\tilde{\mathbf{s}}_3| \leq |\tilde{\mathbf{s}}_1| + |\tilde{\mathbf{s}}_2|$ which shows the bounds on the errors.

Depth estimation will also depend on two vectors \mathbf{v}_1 and \mathbf{v}_2 defined as

$$\mathbf{v}_1 = \hat{\mathbf{s}}_1 - \hat{\mathbf{s}}_1^0, \quad \mathbf{v}_2 = \hat{\mathbf{s}}_2 - \hat{\mathbf{s}}_2^0. \quad (17)$$

With these new vectors it can be proved that

$$\begin{aligned} \tilde{d}_1 &= d_1 - \hat{d}_1 = \frac{|\mathbf{s}_3 - \mathbf{s}_1| - |\hat{\mathbf{s}}_3 - \hat{\mathbf{s}}_1|}{|\mathbf{s}_{31}|} = \frac{|\mathbf{u}_2 - \mathbf{v}_1|}{|\mathbf{s}_{31}|} \\ \tilde{d}_2 &= d_2 - \hat{d}_2 = \frac{|\mathbf{s}_3 - \mathbf{s}_2| - |\hat{\mathbf{s}}_3 - \hat{\mathbf{s}}_2|}{|\mathbf{s}_{32}|} = \frac{|\mathbf{u}_1 - \mathbf{v}_2|}{|\mathbf{s}_{32}|}, \end{aligned} \quad (18)$$

where \mathbf{s}_{31} and \mathbf{s}_{32} are the values obtained using (1) with estimated distances equal to 1.

So expressions of estimation errors have been given using only known data and the errors in the reference positions \tilde{s}_1 and \tilde{s}_2 .

4.2 Errors in the normals to the planes

This subsection shows the analytical expression of the errors as a function of the original errors in the normals. We denote the errors in the angle inputs as $\tilde{\alpha}_1$ and $\tilde{\alpha}_2$. The error in the orientation of the lines that must be intersected can be defined as

$$\tilde{\alpha}_1^* = \tilde{\alpha}_1 + \tilde{\theta}_{31} \quad \tilde{\alpha}_2^* = \tilde{\alpha}_2 + \tilde{\theta}_{32}, \quad (19)$$

where $\tilde{\theta}_{31}$ and $\tilde{\theta}_{32}$ are the errors in the orientation estimations, which can be expressed as functions of $\tilde{\alpha}_1$ and $\tilde{\alpha}_2$ respectively. With these definitions we can build a quadrilateral like the one shown in Fig. 3.

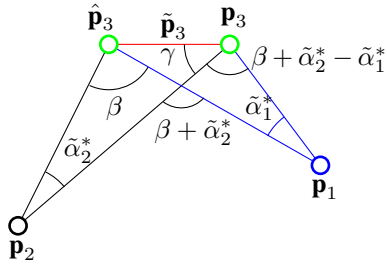


Figure 3. Error obtained in the estimation of s_3 when there are errors in the normals to the planes $\tilde{\alpha}_1$ and $\tilde{\alpha}_2$.

First, in order to obtain the analytical expression of the error we define

$$\beta = \angle([p_2, \hat{p}_3], [p_1, \hat{p}_3]). \quad (20)$$

Looking at Fig. 3 and by using the sinus theorem we obtain the following relations

$$\begin{aligned} \frac{\tilde{p}_3}{\sin \tilde{\alpha}_1^*} &= \frac{|\hat{p}_3 - p_1|}{\sin(\gamma + \beta + \tilde{\alpha}_2^* - \tilde{\alpha}_1^*)} \\ \frac{\tilde{p}_3}{\sin \tilde{\alpha}_2^*} &= \frac{|\hat{p}_3 - p_2|}{\sin \gamma}, \end{aligned} \quad (21)$$

where we still have the problem of the unknown γ . Dividing the two equations of (21) and applying the basic trigonometric rule $\sin(a + b) = \sin(a)\cos(b) + \cos(a)\sin(b)$ we have

$$\sin \gamma A + \cos \gamma B = 0, \quad (22)$$

with A and B

$$\begin{cases} A = 1 - \cos(\beta + \tilde{\alpha}_2^* - \tilde{\alpha}_1^*) \frac{\sin \tilde{\alpha}_2^*}{\sin \tilde{\alpha}_1^*} \frac{|\hat{p}_3 - p_2|}{|\hat{p}_3 - p_1|} \\ B = -\sin(\beta + \tilde{\alpha}_2^* - \tilde{\alpha}_1^*) \frac{\sin \tilde{\alpha}_2^*}{\sin \tilde{\alpha}_1^*} \frac{|\hat{p}_3 - p_2|}{|\hat{p}_3 - p_1|}. \end{cases} \quad (23)$$

Then, $\gamma = \arctan(-B/A)$, function of known data and the error in the normals. Finally, using the value of computed γ we can replace it in eq. (21) obtaining the desired expression of the estimation error \tilde{s}_3 .

In this case we must not forget that there will also be error in the orientation estimation because it depends on the normal but *DEM* will not affect the value of the output orientation.

5 Experimental results

In order to verify our approaches several simulations have been run. We have tried to model the system as close as possible to the reality. In our simulations we represent a room with several walls with different orientations. Each wall has 100 features randomly generated. The camera is simulated using the parameters of the real camera installed in our robots and each image has dimensions of 640x480 pixels. We have done different kind of experiments to show different aspects of our proposal.

5.1 Navigation with pre-learned path

In this scenario the robot makes a teaching phase. While the robot is moving several images are captured and stored to use them later as references. Each image has associated the measurement of the position where it has been taken. In a second step we move the robot along different paths. Using the information of the reference images, the robot estimates its position with high precision. Figure 4 shows an example using *DEM* for some different simple trajectories. In this example the robot has no information about the depth of the planes in the reference images. Figure 5 presents a more complex navigation sequence applying *NOEM*, here there is no information about the normals; the wrong estimations are due to outliers of point intersection of the circumferences. When only one homography can be computed, the sensor uses the *HS* function using as input the last estimation of the normal/distance to the plane. In both examples we have added white gaussian noise to the inputs in order to see the robustness of the techniques.

5.2 Navigation using previous images

In this subsection we show another of the possible applications of these techniques. In this case we capture one image of each plane from the same position. Here we can not use the methods presented because all the images have been taken from the same position. If the distance to the planes introduced in the system is wrong it turns out that the estimated trajectory has error proportional to the depth error (Red line in Fig 6). If the initial position is introduced, then the location of the robot can be rightly estimated using the previous images during the route. This happens although a wrong distance is introduced (Green squares in Fig 6).

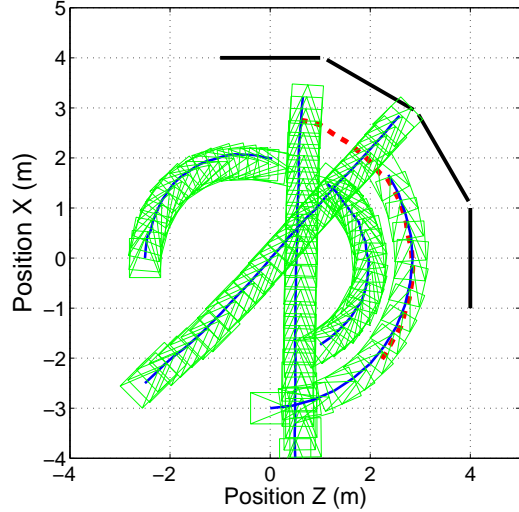


Figure 4. Different paths using *DEM* to obtain metric reconstruction. Black lines represent the reference planes, red dashed line shows the teaching path of the robot, continuous blue lines represent the real paths of the robot and green squares are the estimated locations using *DEM*.

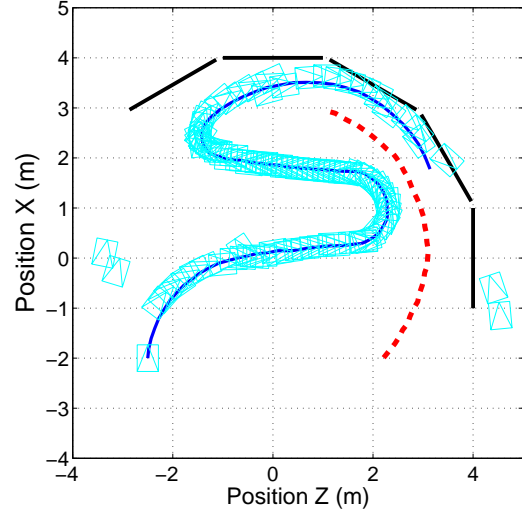


Figure 5. Complex trajectory observed using *NOEM*. In this case we have used blue for the real trajectory and cyan squares to represent the estimated locations. Bad estimations are due to wrong choice of the intersections of the circumferences.

5.3 Combination of both navigations

It is possible to combine both approaches in order to perform harder routes. Fig. 7 shows an example where the robot has a list of images from a pre-learned path. If an obstacle appears in the path (we assume the robot has mechanisms to detect the obstacle), the robot avoids it and the return to the prefixed path. When only one homography can be computed the robot uses images from previous steps (mainly during the obstacle avoidance) and uses them to apply *DEM*. In this case an easy qualitative control has been used to ensure that the robot does not leave the path.

5.4 Error analysis

The last block of simulations pretend to validate the analytical expressions given for the errors. Here we generate three random positions and we take one image of a reference plane from each one. We have added noise to two of the positions (from 0cm to 25cm in each coordinate) and the normal of the planes (15 degrees top in each orientation) and we have estimated the third one using *DEM* equations. The real errors have been measured using all the information available while for the theoretical errors we only have used the data described in the equations (14) to (23). In Fig. 8 we can see this comparison for errors in the reference positions and in Fig. 9 it is possible to observe a surface with the different errors in the orientation of the planes and the obtained errors in the output. Both cases show that the analytical expressions are well calculated.

6 Conclusions and future work

The homography sensor is a precise estimator of the robot location when it is moving indoors. In order to make it more robust we have combined the homographies computed from the images stored in the teaching phase. However, it needs input data which should be as precise as possible. Now it is possible to obtain precise estimations of the position when there is no information about scene depth or orientation. This allows to carry out reactive navigation which is shown by simulations and different paths from those learned during the teaching phase.

We also have given analytical expressions for the errors of one of these methods when there are errors in the inputs. Simulation results confirm our hypothesis. The analytical expressions given here can be useful to model possible uncertainties in the observations in order to develop, for example, a fuzzy navigation algorithm or a control law to move the robot precisely. We are now testing the proposal with real images using conventional cameras.

References

- [1] H. Bay, T. Tuytelaars, and L. V. Gool. Surf: Speeded up robust features. In *European Conference on Computer Vision*, pages 404–417, 2006.
- [2] G. Blanc, Y. Mezouar, and P. Martinet. Indoor navigation of a wheeled mobile robot along visual routes. *IEEE International Conference on Robotics and Automation*, pages 3354–3359, 18–22 April 2005.
- [3] J. Chen, D. M. Dawson, W. E. Dixon, and V. K. Chitrakaran. Navigation function-based visual servo control. *Automatica*, 43(7):1165–1177, 2007.

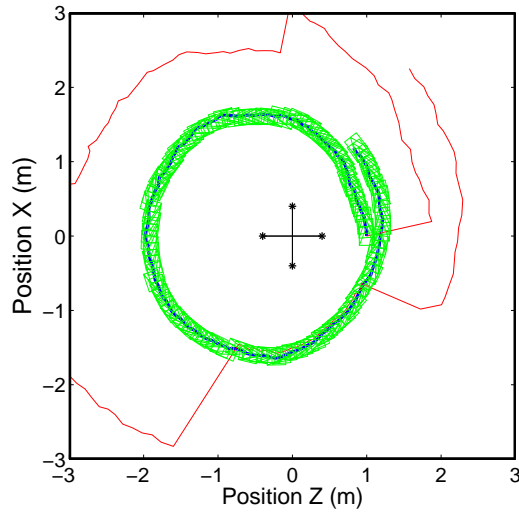


Figure 6. Path in a 4x4 square room. All the reference images are taken from the same position. Some wrong plane depths are introduced in the simulation. If the robot knows its initial position it can perform a reactive navigation. Red line shows the location estimation when no previous images are used. If previous images are used then the estimation is good although the depth of the planes is wrong.

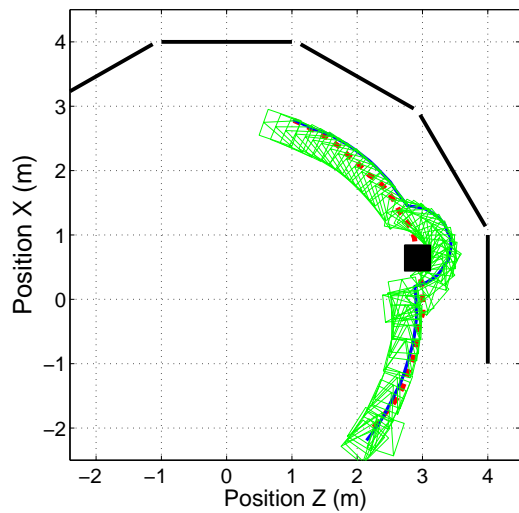


Figure 7. The robot has to repeat a path. When it finds an obstacle it avoids it and then returns to the path without getting lost. In this case it combines the reference images with reactive navigation. In order to follow the path a position-based qualitative control has been used.

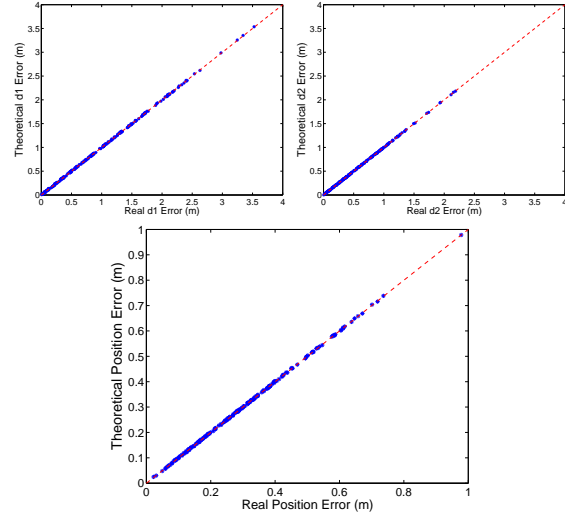


Figure 8. In these graphics it is possible to observe that the theoretical errors coincide with the real errors in the outputs when there are some errors in the reference positions. All the points lay in the line $x = y$ which means that both are equal.

- [4] Z. Chen and S. Birchfield. Qualitative vision-based mobile robot navigation. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 1702–1708, May 2006.
- [5] A. Fusiello. Uncalibrated euclidean reconstruction: A review. *Image and Vision Computing*, 2000.
- [6] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge, 2000.
- [7] D. Lowe. Distinctive image features from scale-invariant keypoints. *Int. Journal of Computer Vision*, 60(2):91–110, 2004.
- [8] E. Malis, F. Chaumette, and S. Boudet. 2 1/2 d visual servoing. *IEEE Transaction on Robotics and Automation*, 15(2):234–246, April 1999.
- [9] 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.
- [10] J. Menudet, J. Becker, T. Fournel, and C. Mennessier. Plane-based camera self-calibration by metric rectification of images. *Image and Vision Computing*, 2007.
- [11] A. Remazeilles and F. Chaumette. Image-based robot navigation from an image memory. *Robot. Auton. Syst.*, 55(4):345–356, 2007.
- [12] C. Sagués and J. Guerrero. Visual correction for mobile robot homing. *Robotics and Autonomous Systems*, 50(1):41–49, 2005.

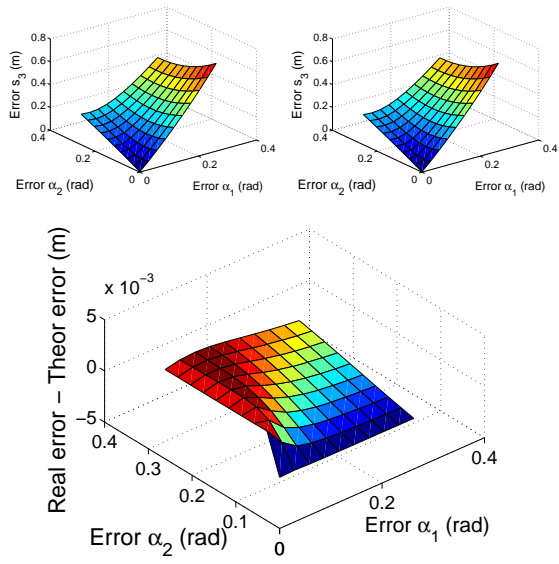


Figure 9. First surface shows the values of the real errors obtained under wrong normal inputs. The second surface shows the values obtained using the analytical expression and the third surface shows the difference of these errors. The correctness of the analytical expression can be observed.