DIIS - I3A
Universidad de Zaragoza
C/ María de Luna num. 1
E-50018 Zaragoza
Spain

Internal Report: 2007-V16

Shortest Path Homography-Based Visual Control for Differential Drive Robots[1]

G. López-Nicolás, C. Sagüés, J.J. Guerrero

If you want to cite this report, please use the following reference instead:
Shortest Path Homography-Based Visual Control for Differential Drive Robots, G. López-Nicolás, C. Sagüés, J.J. Guerrero *Vision Systems - Applications*, Editors: G. Obinata and A. Duta, ISBN 978-3-902613-01-1, Cap. 30, pages 583-596, 2007.

[1] This work was supported by the projects DPI2006-07928, IST-1-045062-URUS-STP.

Shortest Path Homography-Based Visual Control for Differential Drive Robots

G. López-Nicolás, C. Sagüés, and J.J. Guerrero*

Universidad de Zaragoza, DIIS - I3A.
Spain

1 Introduction

It is generally accepted that machine vision is one of the most important sensory modalities for navigation purposes. Visual control, also called visual servoing, is a very extensive and mature field of research where many important contributions have been presented in the last decade [Malis et al., 1999, Corke and Hutchinson, 2001, Conticelli and Allotta, 2001, Tsakiris et al., 1998, Ma et al., 1999]. Two interesting surveys on this topic are [DeSouza and Kak, 2002] and [Hutchinson et al., 1996]. In this work we present a new visual servoing approach for mobile robots with a fixed monocular system on board. The idea of visual servoing is used here in the sense of homing, where the desired robot position is defined by a target image taken at that position. Using the images taken during the navigation the robot is led to the target.

A traditional approach is to perform the motion by using the epipolar geometry [Basri et al., 1999, Rives, 2000, López-Nicolás et al., 2006]. These approaches have as drawback that the estimation of the epipolar geometry becomes ill conditioned with short baseline or planar scenes, which are usual in human environments. A natural way to overcome this problem is using the homography model. In [Malis and Chaumette, 2000] it is proposed a method based on the estimation of the homography matrix related to a virtual plane attached to an object. This method provides a more stable estimation when the epipolar geometry degenerates. In [Benhimane et al., 2005] it is presented a visual tracking system for car platooning by estimating the homography between a selected reference template attached to the leading vehicle. A significant issue with monocular camera-based vision systems is the lack of depth information. In [Fang et al., 2005] it is proposed the asymptotic regulation of the position and orientation of a mobile robot by exploiting homography-based visual servo control strategies, where the unknown time-varying depth information is related to a constant depth-related parameter.

These homography-based methods usually require the homography decomposition, which is not a trivial issue. Two examples of approaches which do not use the decomposition of the homography are [Sagüés and Guerrero, 2005] which is based on a 2D homography and [Benhimane and Malis, 2006] which presents an uncalibrated approach for manipulators. We present a novel homography-based

* This work was supported by projects DPI2006-07928, IST-1-045062-URUS-STP.

approach by performing the control directly on the elements of the homography matrix. This approach, denoted as "*Shortest Path Control*", is based on the design of a specific robot trajectory which consists in following a straight line towards the target. This motion planning allows to define a control law decoupling rotation and translation by using the homography elements. This approach needs neither the homography decomposition nor depth estimation. In this work we have developed three similar methods based on the particular selection of the homography elements. Each method is suitable for different situations.

The chapter is divided as follows, Section 2 presents the homography model developing its elements as a function of the system parameters to be used in the design of the controllers. Section 3 presents the *Shortest Path Control* with three different methods based on the elements of the homography. Sections 4 and 5 present the stability analysis of the controllers and the experimental results respectively. Section 6 gives the conclusions.

2 Homography Based Model

The general pinhole camera model considers a calibration matrix defined as

$$\mathbf{K} = \begin{bmatrix} \alpha_x & s & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{bmatrix} , \tag{1}$$

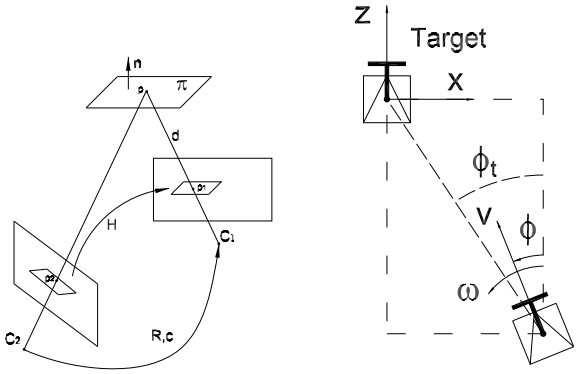
where α_x and α_y are the focal length of the camera in pixel units in the x and y directions respectively; s is the skew parameter and (x_0, y_0) are the coordinates of the principal point. We have that $\alpha_x = f\, m_x$ and $\alpha_y = f\, m_y$, where f is the focal length and m_x, m_y are the number of pixels per distance unit. In practice, we assume that the principal point is in the centre of the image ($x_0 = 0, y_0 = 0$) and that there is no skew ($s = 0$).

A 3D point in the world can be represented in the projective plane with homogeneous coordinates as $\mathbf{p} = (x, y, 1)^T$. A projective transformation $\mathbf{H}$ exists from matched points belonging to a plane in such a way that $\mathbf{p}_2 = \mathbf{H}\,\mathbf{p}_1$. The homography between the current and target image can be computed from the matched points, and a robust method like RANSAC should be used to consider the existence of outliers [Hartley and Zisserman, 2004]. Taking advantage of the planar motion constraint, the homography can be computed from three correspondences instead of four, reducing the processing time.

Let us suppose two images obtained with the same camera whose projection matrixes in a common reference system are $\mathbf{P}_1 = \mathbf{K}[\mathbf{I}|\mathbf{0}]$ and $\mathbf{P}_2 = \mathbf{K}[\mathbf{R}| - \mathbf{R}\mathbf{c}]$, being $\mathbf{R}$ the camera rotation and $\mathbf{c}$ the translation between the optical centres of the two cameras. A homography $\mathbf{H}$ can be related to camera motion (Fig. 1a) in such a way that

$$\mathbf{H} = \mathbf{K}\,(\mathbf{R} - \mathbf{t}\,\frac{\mathbf{n}^T}{d})\,\mathbf{K}^{-1} = \mathbf{K}\,(\mathbf{R} + \mathbf{R}\mathbf{c}\,\frac{\mathbf{n}^T}{d})\,\mathbf{K}^{-1} = \mathbf{K}\mathbf{R}\,(\mathbf{I} + \mathbf{c}\,\frac{\mathbf{n}^T}{d})\,\mathbf{K}^{-1} \tag{2}$$

where $\mathbf{n} = (n_x, n_y, n_z)^T$ is the normal to the plane that generates the homography and d is the distance between the plane and the origin of the global reference.



(a) (b)

Fig. 1. (a) Homography from a plane between two views. (b) Coordinate system.

We consider a mobile robot in planar motion (Fig. 1b). In this case the robot position is defined by the state vector (x, z, ϕ) and the planar motion constraint gives:

$$\mathbf{R} = \begin{bmatrix} \cos\phi & 0 & \sin\phi \\ 0 & 1 & 0 \\ -\sin\phi & 0 & \cos\phi \end{bmatrix}, \quad \mathbf{c} = (x,\, 0,\, z)^T . \tag{3}$$

Taking this into account, the homography corresponding to a planar motion scheme can be written as

$$\mathbf{H} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ 0 & 1 & 0 \\ h_{31} & h_{32} & h_{33} \end{bmatrix}, \tag{4}$$

The second row of the matrix will be ignored in the design of the control law as it does not give useful information. Developing expression (2) we obtain the homography elements as a function of the parameters involved:

$$\begin{aligned} h_{11} &= \cos\phi + (x\cos\phi + z\sin\phi)\tfrac{n_x}{d} \\ h_{12} &= \tfrac{\alpha_x}{\alpha_y}(x\cos\phi + z\sin\phi)\tfrac{n_y}{d} \\ h_{13} &= \alpha_x\left(\sin\phi + (x\cos\phi + z\sin\phi)\tfrac{n_z}{d}\right) \\ h_{31} &= \tfrac{1}{\alpha_x}\left(-\sin\phi + (-x\sin\phi + z\cos\phi)\tfrac{n_x}{d}\right) \\ h_{32} &= \tfrac{1}{\alpha_y}(-x\sin\phi + z\cos\phi)\tfrac{n_y}{d} \\ h_{33} &= \cos\phi + (-x\sin\phi + z\cos\phi)\tfrac{n_z}{d} \end{aligned} \tag{5}$$

The analysis of these homography elements will lead to the control law design. After computing the homography from the image point matches it has to be

normalized. We normalize by dividing $\mathbf{H}/h_{22}$, given that h_{22} is never zero due to the planar motion constraint.

3 Visual Servoing with *Shortest Path Control*

In this Section the *Shortest Path Control* is presented. The control law design is directly based on the homography elements. Given that our system has two variables to be controlled (the velocities v and ω), we need at least two parameters of the homography to define the control law. Several possibilities appear depending on which homography elements are selected. In our approach we have developed three similar methods which are suitable for different situations. In the experimental results we show the performance of these methods as the calibration or the scene change.

Let us suppose the nonholonomic differential kinematics to be expressed in a general way as

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}) \tag{6}$$

where $\mathbf{x} = (x, z, \phi)^T$ denotes the state vector and $\mathbf{u} = (v, \omega)^T$ the input vector. The particular nonholonomic differential kinematics of the robot expressed in state space form as a function of the translation and rotation robot velocities (v, ω) is:

$$\begin{pmatrix} \dot{x} \\ \dot{z} \\ \dot{\phi} \end{pmatrix} = \begin{pmatrix} -\sin\phi \\ \cos\phi \\ 0 \end{pmatrix} v + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \omega \tag{7}$$

In the *Shortest Path Control* approach, we propose decoupling the motion, rotation and translation, by following a specific trajectory. Then, we design a navigation scheme in such a way that the robot can correct rotation and translation in a decoupled way. The resulting path of this motion is shown in Fig. 2.

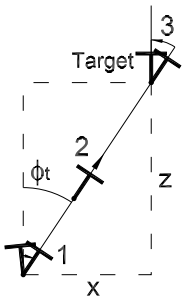


Fig. 2. Motion trajectory of the robot consisting in three steps.

The motion can be divided in three sequential steps. In the first step the robot rotates until the camera points to the target position. Then, the robot

performs a straight translation in the second step until the target position is reached up to a rotation. Finally, the orientation is corrected in the third step. The key point is to establish what conditions have to be held during each phase of the navigation. When the motion starts, the initial homography is the general case (5). It can be seen in Fig. 2 that during the second step the robot moves in a straight line with a constant angle respect the global reference ($\phi = \phi_t$). From our reference system we can obtain the geometrical expression $x = -z \tan \phi_t$. Using this expression in (5) we obtain the particular form of the homography that is held during the straight motion of the second step:

$$\mathbf{H}(\phi = \phi_t) = \begin{bmatrix} \cos\phi_t & 0 & \alpha_x \sin\phi_t \\ 0 & 1 & 0 \\ \frac{1}{\alpha_x}\left(-\sin\phi_t + \frac{z}{\cos\phi_t}\frac{n_x}{d}\right) & \frac{1}{\alpha_y}\frac{z}{\cos\phi_t}\frac{n_y}{d} & \cos\phi_t + \frac{z}{\cos\phi_t}\frac{n_z}{d} \end{bmatrix} \quad (8)$$

At the end of the second step the robot has an orientation error and no translation error ($x = 0$, $z = 0$, $\phi = \phi_t$). Taking this into account, the homography matrix that results at the end of the second step (i.e. in the target position up to orientation error) is:

$$\mathbf{H}(x = 0, z = 0, \phi = \phi_t) = \begin{bmatrix} \cos\phi_t & 0 & \alpha_x \sin\phi_t \\ 0 & 1 & 0 \\ \frac{-\sin\phi_t}{\alpha_x} & 0 & \cos\phi_t \end{bmatrix} \quad (9)$$

This previous expression also implies that $det(\mathbf{H}) = 1$. Finally, at the end of the navigation, when the robot reaches the target pose with the desired orientation the homography will be the identity matrix,

$$\mathbf{H}(x = 0, z = 0, \phi = 0) = \mathbf{I}\,. \quad (10)$$

The particular expressions of the homography just deduced are related graphically with its corresponding positions in Fig. 3. It can be seen that the goal of each step is to move the robot having as reference the next desired expression of the homography.

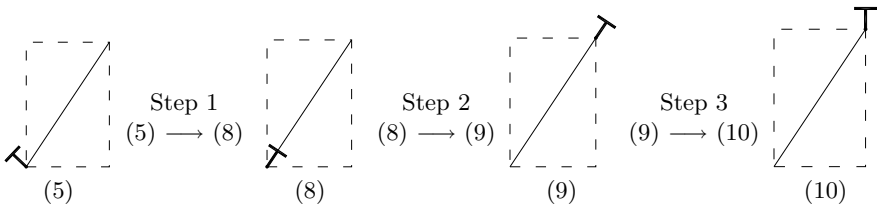


Fig. 3. The number below each figure denotes the equation of the homography that holds in that position. In each step, the numbers give the homography equations at the start and at the end of the step.

Now we briefly introduce the expressions used to define the controllers of the three different methods of the *Shortest Path Control*. These are detailed

in the following subsections. From the previous particular expressions of the homography, we can define the conditions that will be used in each step of the navigation to drive the robot. In the first step we want to reach the orientation $\phi = \phi_t$, where the robot points to the target. The forward velocity is set to zero ($v = 0$) and from (8) we could use h_{11}, h_{12} or h_{13} to set the angular velocity of the robot in a proportional control:

$$\omega = k_\omega(h_{11} - \cos\phi_t) \tag{11}$$

$$\omega = k_\omega h_{12} \tag{12}$$

$$\omega = k_\omega(h_{13} - \alpha_x \sin\phi_t) \tag{13}$$

In this step we have rejected elements h_{31}, h_{32} and h_{33} because they require knowledge about the plane and the robot position, which are unknown. Each one of these expressions (11), (12) or (13) can be used to correct rotation in the first step. The selection of the expressions for each of the three methods depending on the calibration hypothesis is explained below. In method I camera calibration is supposed to be known, while in Method II and III no specific calibration is required.

Once the orientation ϕ_t is gained, the second step aims to get translation to the target equal to zero ($x = z = 0$), keeping the orientation constant during the motion ($\phi = \phi_t$). In this case we could use the parameters h_{31}, h_{32} or h_{33} from (9) to set the robot velocity as

$$v = k_v(h_{31} + \frac{\sin\phi_t}{\alpha_x}) \tag{14}$$

$$v = k_v h_{32} \tag{15}$$

$$v = k_v(h_{33} - \cos\phi_t) \tag{16}$$

In this second step we have rejected elements h_{11}, h_{12} and h_{13} for the correction of v because the value of these elements is constant during this step. Any of the expressions (14), (15) or (16) can be used to compute v during this step. Odometry drift or image noise appear in real situations, so the orientation is corrected to avoid possible errors. Thus, in the three methods the rotation during second step is corrected respectively with the same control of the first step.

In the last step the robot has zero translation error and only needs to perform a rotation in order to reach the target orientation,

$$\omega = k_\omega(h_{ij} - 1)\ with\ (i, j = 1, 3),\ (i = j) \tag{17}$$

$$\omega = k_\omega h_{ij}\ with(j = 1, 2, 3),\ (i \neq j) \tag{18}$$

Then, the velocity is set to zero in this step ($v = 0$) and the rotation can be corrected with expressions of (17) or (18). We have selected $\omega = -k_\omega h_{13}$ for the three methods because of the robustness to noise of h_{13} with respect to the rest of the homography elements. Experimental results presented support this decision.

The control loop of the scheme presented is shown in the diagram of Fig. 4. An image in the current position is taken at each loop of the control. The homography that links it with the target image is computed from the feature matching. Using the homography, the control performs the three steps. When the homography-based control loop finishes, the robot is in the target position, the current and the target images are the same, and the homography is the identity matrix. Next, the details of the three methods of the *Shortest Path Control* for visual servoing based on homographies for mobile robots are presented in detail.

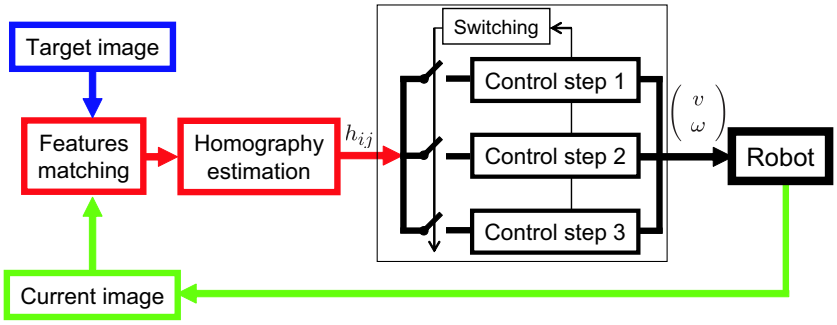


Fig. 4. Diagram of the control loop.

3.1 Method I: Calibrated Method

In this method we suppose that the calibration matrix of the camera is known, and therefore, the value of the focal length α_x is given. In the first step v is set to zero while the angular velocity could be corrected with (11) or (13), needing the value ϕ_t. This approach is based on the key value ϕ_t, but this value is initially unknown. From (8) we have that $h_{11} = \cos\phi_t$ and $h_{13} = \alpha_x \sin\phi_t$. Taking this into account, we can obtain the next equation, which is true when $\phi = \phi_t$,

$$h_{11}^2 + \frac{h_{13}^2}{\alpha_x^2} = 1 \tag{19}$$

Using this expression we do not need to know the value of ϕ_t to correct the orientation in the first step, and this is corrected until (19) is satisfied. In step two, the orientation is corrected with the same expression to take into account odometry drift or noise. The velocity v in the second step is corrected using (16) which is combined with h_{11} from (9) to remove the unknown parameter ϕ_t from the expression of the control. Third step is based on (17). Then, we define the

Method I as

$$Method\ I \begin{cases} Step\ 1: \ v = 0\,,\ \omega = -k_\omega(h_{11}^2 + \frac{h_{13}^2}{\alpha_x^2} - 1)\,. \\ Step\ 2: \ v = -k_v(h_{11} - h_{33})\,,\ \omega = -k_\omega(h_{11}^2 + \frac{h_{13}^2}{\alpha_x^2} - 1)\,. \\ Step\ 3: \ v = 0\,,\ \omega = -k_\omega h_{13}\,. \end{cases} \quad (20)$$

where k_ω and k_v are the control gains.

We avoid the use of the parameter ϕ_t in the velocity v of the second step by using the value of h_{11} from (9) as previously explained. In any case ϕ_t could be computed easily when the first step is finished from (11) or (13). This method needs to know the calibration of the camera (parameter α_x) and this is its main drawback. The next two methods proposed work without knowing this parameter and they have shown to be independent of the focal length.

3.2 Method II: Uncalibrated Method

The previous method is calibrated. In a system, the need of calibration means disadvantages in terms of maintenance cost, robustness and adaptability. In Method II the calibration camera is considered to be unknown, which has many advantages in practice. We can define the control scheme of the Method II selecting expressions where the calibration parameters do not appear explicitly. These expressions are (12), (15) and (17). Then, the control is defined as

$$Method\ II \begin{cases} Step\ 1: \ v = 0\,,\ \omega = -k_\omega h_{12}\,. \\ Step\ 2: \ v = -k_v h_{32}\,,\ \omega = -k_\omega h_{12}\,. \\ Step\ 3: \ v = 0\,,\ \omega = -k_\omega h_{13}\,. \end{cases} \quad (21)$$

where k_ω and k_v are the control gains. With this method the robot is controlled by using a camera without specific calibration; although we assume that the principal point is in the centre of the image, this is a good supposition in practise. Method II requires the plane inducing the homography not to be vertical respect our reference because it is needed $n_y \neq 0$. This is due to the direct dependence of the parameters used from the homography to n_y. This could be a problem since human environments are usually full of vertical planes (walls). In any case the method works if we guarantee that vertical planes are not used, for example constraining to the floor [Liang and Pears, 2002] or the ceiling plane [Blanc et al., 2005].

3.3 Method III: Method with Parallax

The previous method works without specific calibration, but it requires the scene homography plane not to be vertical and this could be a problem in man-made environments, usually full of vertical planes. Method III uses the concept of parallax relative to a plane and overcomes the problem of vertical planes. Using the parallax [Hartley and Zisserman, 2004] the epipole in the current image can be easily obtained from a homography $\mathbf{H}$ and two points not belonging to its

plane. In the first step of Method III the objective is to get orientation $\phi = \phi_t$. In this position the robot points to the target, so the camera centre of the target is projected to (x_0, y_0) in the current image and then $\mathbf{e}_c = (0, 0)$. Given that the robot moves in a planar surface we only need the x-coordinate of the epipole (e_{cx}). Then we define the correction of the orientation in step 1 and step 2 with a proportional control to e_{cx}. Once $e_{cx} = 0$ the robot is pointing to the target position. The other expressions of the control are obtained in a similar way to the previous methods using (16) and (17). Then, we define the scheme of Method III as

$$Method\ III \begin{cases} Step\ 1: \ v = 0\,,\ \omega = -k_\omega e_{cx}\,. \\ Step\ 2: \ v = -k_v(h_{11} - h_{33})\,,\ \omega = -k_\omega e_{cx}\,. \\ Step\ 3: \ v = 0\,,\ \omega = -k_\omega h_{13}\,. \end{cases} \tag{22}$$

When the robot is close to the target position and the translation is nearly zero, all the points in the scene can be related by the homography. In this situation the parallax is not useful to correct the orientation. Before this happen we change the orientation control at the end of step 2 to the expression (11). This expression needs the value of ϕ_t, which can be computed previously with the same equation while the rotation is corrected with the parallax procedure. Here, we use neither expression (15) because vertical planes can be easily found in human environments nor expression (19) because it needs specific calibration. We can detect easily when the parallax is not useful to work with by measuring the parallax of the points not belonging to the plane of the homography. If the result is under a threshold, the parallax procedure is not used anymore. In the simulations presented with this approach the threshold is set to 5 pixels.

In the three methods presented the homography is not decomposed, and neither the robot coordinates nor the normal of the plane are computed. This approach requires the selection of the signs of some of the control gains depending on where is the initial robot position and what is the orientation of the plane detected. This can be easily done by taking advantage of the parallax relative to the plane by computing it once at the start. Thus, the sign of the gains is easily determined.

4 Stability Analysis

We define the common Lyapunov function expressing the robot position in polar coordinates $(r(t),\ \theta(t),\ \phi(t))$, with the reference origin in the target and θ positive from z-axis anticlockwise, as

$$V = V_r + V_\theta + V_\phi = \frac{(r - r^{G_i})^2}{2} + \frac{(\theta - \theta^{Gi})^2}{2} + \frac{(\phi - \phi^{G_i})^2}{2}\,. \tag{23}$$

This is a positive definite function, where r^{G_i}, θ^{Gi} and ϕ^{Gi} denote the desired value of the parameter in the subgoal position for each step $(i = 1, 2, 3)$. Due to the designed path, the value of θ is constant during the navigation. Although in the case of noisy data the value of θ could vary, it does not affect the control,

because the path is defined towards the target independently of the value of θ, thus $V_\theta = 0$. After differentiating we obtain:

$$\dot{V} = \dot{V}_r + \dot{V}_\phi = (r - r^{Gi})\, v\, \cos(\phi - \theta) + (\phi - \phi^{Gi})\, \omega\ . \tag{24}$$

We analyze the derivative Lyapunov candidate function in each step to show it is strictly negative. This analysis is valid whether if the goal is behind or in front of the initial position.

Step 1 Here the robot performs a rotation with $v = 0$. Thus, we only need to consider $\dot{V} = \dot{V}_\phi$. The desired orientation is $\phi^{G_1} = \phi_t$. $\dot{V}_\phi < 0$ is guaranteed if $(\phi - \phi^{G_1}) > 0$ and then $\omega < 0$; or else, if $(\phi - \phi^{G_1}) < 0$ and then $\omega > 0$. In Method I and II, the sign of ω is guaranteed to be correct, given that the sign of k_ω is selected as previously explained. In Method III, $\omega = -k_\omega e_{cx}$ and, when $(\phi - \phi^{G_1}) > 0$ then $e_{cx} > 0$ and $\omega < 0$, or $e_{cx} < 0$ and $\omega > 0$ when $(\phi - \phi^{G_1}) < 0$. Therefore $\dot{V} < 0$.

Step 2 In this step the robot moves towards the target in a straight line path and we have $\dot{V} = \dot{V}_r + \dot{V}_\phi$. The sign of $(r - r^{G_2})$ is always positive. Then, with $\cos(\phi - \theta) < 0$ we have $v > 0$ and with $\cos(\phi - \theta) > 0$ we have $v < 0$. In Method II, the sign of v is guaranteed to be correct, given that the sign of k_v is properly selected. In Method I and III, the velocity given by the control and with (8) is $v = k_v z n_z/(d \cos\phi_t)$, which gives the expected signs. Therefore $\dot{V}_r < 0$. With $\dot{V}_\phi$ we have the same reasoning of step 1.

Step 3 Similar to the reasoning of step 1, in this case, the sign of ω can be easily checked taking into account that $\phi^{G_3} = 0$ and $h_{13} = \alpha_x \sin\phi_t$. Therefore $\dot{V} < 0$.

So, we have shown that $\dot{V} < 0$ for the controllers of the three methods. We have also asymptotic stability given that $\dot{V}$ is negative definite in all the steps.

5 Experimental Results

Several experiments have been carried out with the controllers of the three methods presented by using virtual data. The simulated data is obtained by generating a virtual planar scene consisting of a distribution of random 3D points. The scene is projected to the image plane using a virtual camera, the size of the images is 640×480 pixels. In each loop of the control, the homography between the current and target image is computed from the matched points and the control law send the velocities (v, ω) to the robot. In the experiments, we assume that the camera is centred on the robot pointing forwards. Figure 5 shows the resulting path from different initial positions. The target is placed in $(x(\mathrm{m}), z(\mathrm{m}), \phi(\mathrm{deg})) = (0, 0, 0^\circ)$. The different initial positions behind the target are: $(-3, -10, -30^\circ)$, $(0, -8, -40^\circ)$ and $(6, -6, 0^\circ)$. The results also show that

the method works properly when the target is behind the initial robot position, moving the robot backwards in that case. The different initial positions used in this case are: $(-6, 4, 20°)$, $(6, 8, 10°)$ and $(5, 2, -50°)$.

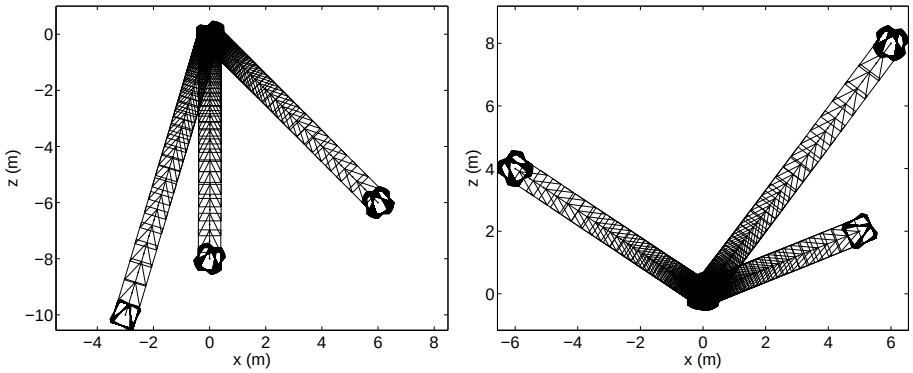
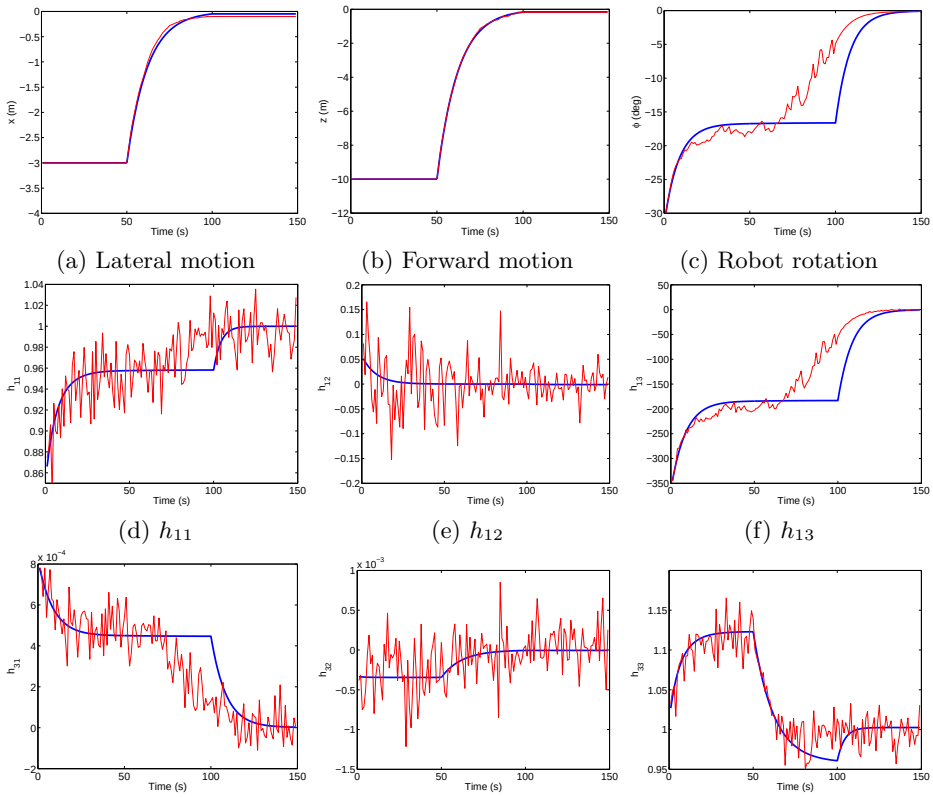


Fig. 5. Simulations with target position at $(0, 0, 0°)$ and different initial positions.

The performance of the three methods is exactly the same when using perfect data and quite similar when there is image noise. In Fig. 6 two simulations are compared, one without noise, and the other, adding white noise to the image points with a standard deviation of $\sigma = 1$ pixel using Method III. The evolution along time of the robot position and the homography elements is drawn.

We have tested the controllers with odometry drift and with different values of image noise. The first row of Fig. 7 shows the resulting evolution of the robot position when there is odometry drift in rotation of 1 deg/m. As it can be seen the controllers can cope properly with the drift error. Simulations with each method have been carried out using different levels of image noise. The results are shown in the second row of Fig. 7 and it can be seen that the methods converge properly in spite of image noise.

The control law of Method I needs the calibration parameter α_x of the camera whereas Method II and III do not use it. In Fig. 8 we show the performance of the control to calibration errors. The value of the focal length of the controllers is fixed to $f = 6$ mm while its real value is modified to see the final position error obtained for each Method, (first row of Fig. 8). Besides, we have assumed that the principal point is in the centre of the image. In the second row of Fig. 8, the value of x_0 used in the controllers is supposed to be zero while its real value is changed. Performance of Method I is sensitive to calibration errors as expected, this is because this control law is related directly with α_x and depends highly on its accuracy. The simulations show that Method II works properly in spite of calibration errors. Finally, results using Method III show that a rough calibration is enough for the convergence, because it is robust to focal length inaccuracy and it is only affected by calibration errors in the principal point.



(a) Lateral motion (b) Forward motion (c) Robot rotation

(d) h_{11} (e) h_{12} (f) h_{13}

(g) h_{31} (h) h_{32} (i) h_{33}

Fig. 6. Simulation without noise (thick line) and with image white noise of $\sigma = 1$ pixel (thin line). The initial position is $(x, z, \phi) = (-3, -10, -30^\circ)$ and the target $(0, 0, 0^\circ)$.

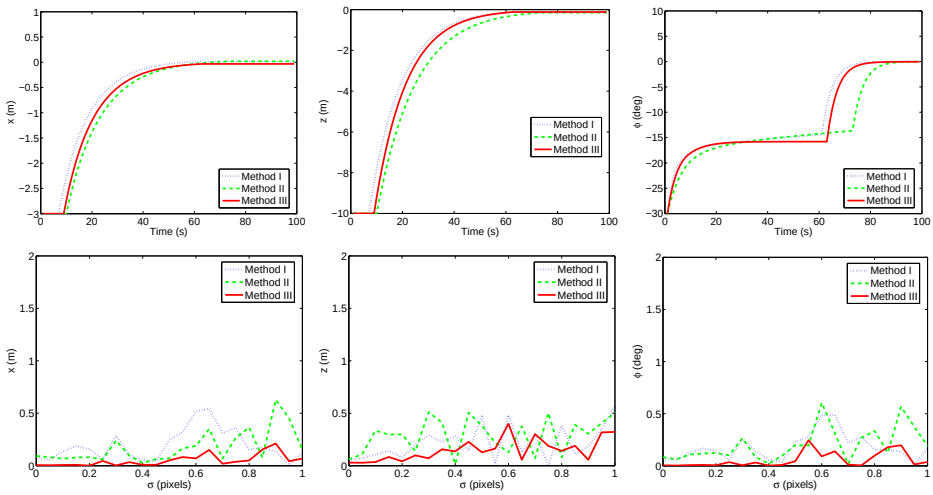


Fig. 7. (First row) Simulations with odometry drift of 1 deg/m. The evolution of one simulation in x, z and ϕ is shown for each method. (Second row) Final error of different simulations varying the image noise.

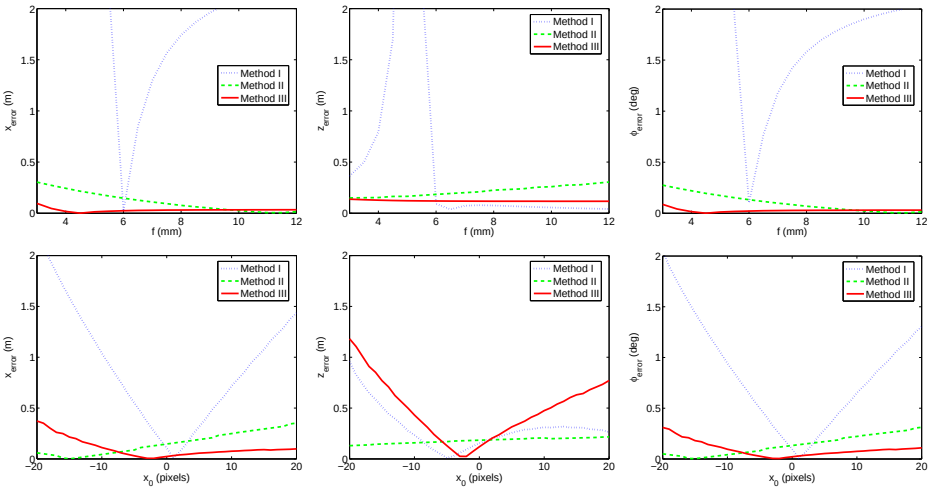


Fig. 8. Final error for each method in x, z and ϕ varying the focal length (first row) and varying the principal point coordinates (second row).

The performance of the methods can be spoiled in some cases by the particular plane that generates the homography. Simulations using different planes are presented in Table 1. The planes are defined by the normal vector $\mathbf{n} = (n_x, n_y, n_z)^T$, and a list of unitary normal vectors is selected to carried out the simulations with $\|n\| = 1$. The final error obtained with each method is shown. The initial position is $(-3, -10, -30°)$ and the target is $(0, 0, 0°)$. The results show that Method I and III need $n_z \neq 0$ to work properly. On the other hand, Method II needs $n_y \neq 0$. This is because the Methods are directly related with these parameters of $\mathbf{n}$. Vertical planes are usually common in human environments; besides, in our monocular system, planes in front of the robot with dominant n_z will be detected more easily. Methods I and III work properly in this case. If we constraint the homography plane detected to be the floor or the ceiling (any plane with $n_y \neq 0$ is enough) the Method II will also work properly.

Table 1. Final error for each method in x(m), z(m) and ϕ(deg) varying the normal of the plane that generates the homography: $\mathbf{n} = (n_x, n_y, n_z)^T$.

n			Method I			Method II			Method III		
n_x	n_y	n_z	x	z	ϕ	x	z	ϕ	x	z	ϕ
0	0	-1.00	0	0	-0.09	-3.00	-10.00	-3.12	0	0	-0.09
-0.20	0.57	-0.80	0.03	-0.00	-0.09	-0.00	-0.00	-0.09	0	0	-0.09
-0.40	0.69	-0.60	-0.00	-0.00	-0.09	-0.00	-0.00	-0.09	-0.00	-0.00	-0.09
-0.60	0.69	-0.40	-0.00	-0.01	-0.09	-0.00	-0.00	-0.09	-0.00	-0.01	-0.09
-0.80	0.57	-0.20	-0.10	-0.34	-0.03	-0.00	-0.00	-0.09	-0.10	-0.34	-0.03
-1.00	0	0	-3.00	-10.00	0	-3.00	-10.00	0	-3.00	-10.00	0
1.00	0	0	-3.00	-10.00	0	-3.00	-10.00	0	-3.00	-10.00	0
0.98	-0.20	0	-3.00	-10.00	0	-0.15	-0.62	0	-3.00	-10.00	0
0.92	-0.40	0	-3.00	-10.00	0	-0.01	-0.04	-0.09	-3.00	-10.00	0
0.80	-0.60	0	-3.00	-10.00	0	-0.00	-0.00	-0.09	-3.00	-10.00	0
0.60	-0.80	0	-3.00	-10.00	0	0	-0.00	-0.09	-3.00	-10.00	0
0	-1.00	0	-3.00	-10.00	0	0	0	-0.09	-3.00	-10.00	0
0	-1.00	0	-3.00	-10.00	0	0	0	-0.09	-3.00	-10.00	0
0.57	-0.80	-0.20	-0.10	-0.34	-0.03	0	-0.00	-0.09	-0.10	-0.34	-0.03
0.69	-0.60	-0.40	-0.00	-0.01	-0.09	-0.00	-0.00	-0.09	-0.00	-0.01	-0.09
0.69	-0.40	-0.60	-0.00	-0.00	-0.09	-0.01	-0.04	-0.10	-0.00	-0.00	-0.09
0.57	-0.20	-0.80	0	0	-0.09	-0.15	-0.62	-0.15	0	0	-0.09
0	0	-1.00	0	0	-0.09	-3.00	-10.00	-3.12	0	0	-0.09

6 Conclusions

We have presented a new homography-based approach for visual control of mobile robots. The control design is directly based on the homography elements and deals with the motion constraints of the differential drive vehicle. In our approach, called *Shortest Path Control*, the motion is designed to follow a straight

line path. Taking advantage of this specific trajectory we have proposed a control law decoupling rotation and translation. Three different methods have been designed by choosing different homography elements. Their performance depends on the conditions of the plane or the calibration. The methods use neither the homography decomposition nor any measure of the 3D scene. Simulations shows the performance of the methods with odometry drift, image noise and calibration errors. Also, the influence of the plane that generates the homography is studied.

References

[Basri et al., 1999] Basri, R., Rivlin, E., and Shimshoni, I. (1999). Visual homing: Surfing on the epipoles. *International Journal of Computer Vision*, 33(2):117–137.

[Benhimane and Malis, 2006] Benhimane, S. and Malis, E. (2006). Homography-based 2d visual servoing. *IEEE International Conference on Robotics and Automation*, pages 2397–2402.

[Benhimane et al., 2005] Benhimane, S., Malis, E., Rives, P., and Azinheira, J. R. (2005). Vision-based control for car platooning using homography decomposition. In *IEEE International Conference on Robotics and Automation, Barcelona, Spain*, pages 2173–2178.

[Blanc et al., 2005] Blanc, G., Mezouar, Y., and Martinet, P. (2005). Indoor navigation of a wheeled mobile robot along visual routes. In *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA'05*, pages 3365–3370.

[Conticelli and Allotta, 2001] Conticelli, F. and Allotta, B. (2001). Nonlinear controllability and stability analysis of adaptive image-based systems. *IEEE Transactions on Robotics and Automation*, 17(2):208–214.

[Corke and Hutchinson, 2001] Corke, P. I. and Hutchinson, S. A. (2001). A new partitioned approach to image-based visual servo control. *IEEE Transactions on Robotics and Automation*, 17(4):507–515.

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

[Fang et al., 2005] Fang, Y., Dixon, W. E., Dawson, D. M., and Chawda, P. (2005). Homography-based visual servo regulation of mobile robots. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 35(5):1041–1050.

[Hartley and Zisserman, 2004] Hartley, R. I. and Zisserman, A. (2004). *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518.

[Hutchinson et al., 1996] Hutchinson, S., Hager, G., and Corke, P. (1996). A tutorial on visual servo control. *IEEE Transactions on Robotics and Automation*, 12(5):651–670.

[Liang and Pears, 2002] Liang, B. and Pears, N. (2002). Visual navigation using planar homographies. In *IEEE Conference on Robotics and Automation*, pages 205–210.

[López-Nicolás et al., 2006] López-Nicolás, G., Sagüés, C., Guerrero, J., Kragic, D., and Jensfelt, P. (2006). Nonholonomic epipolar visual servoing. *IEEE International Conference on Robotics and Automation*, pages 2378–2384.

[Ma et al., 1999] Ma, Y., Kosecka, J., and Sastry, S. (1999). Vision guided navigation for a nonholonomic mobile robot. *IEEE Transactions on Robotics and Automation*, 15(3):521–537.

[Malis and Chaumette, 2000] Malis, E. and Chaumette, F. (2000). 2 1/2 D visual servoing with respect to unknown objects through a new estimation scheme of camera displacement. *International Journal of Computer Vision*, 37(1):79–97.

[Malis et al., 1999] Malis, E., Chaumette, F., and Boudet, S. (1999). 2 1/2 D visual servoing. *IEEE Transactions on Robotics and Automation*, 15(2):234–246.

[Rives, 2000] Rives, P. (2000). Visual servoing based on epipolar geometry. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 1, pages 602–607.

[Sagüés and Guerrero, 2005] Sagüés, C. and Guerrero, J. (2005). Visual correction for mobile robot homing. *Robotics and Autonomous Systems*, 50(1):41–49.

[Tsakiris et al., 1998] Tsakiris, D., Rives, P., and Samson, C. (1998). Extending visual servoing techniques to nonholonomic mobile robots. In G. Hager, D. K. and Morse, S., editors, *The Confluence of Vision and Control*, Lecture Notes in Control and Information Systems (LNCIS). Springer-Verlag.