

---

## Survey of Visual Servoing Control Schemes for Mobile Robot Navigation

\*Hend M. Hafez<sup>1</sup>, Mohammed A.R. Marey<sup>2</sup>, Farid A. Tolbah<sup>3</sup> and Magdy M. Abdelhameed<sup>4</sup>

<sup>1</sup>Mechatronics Engineering Department, High Institute of Engineering – 6th October University, Cairo, Egypt

<sup>2</sup>Faculty of Computer and Information Sciences, Ain Shams University, Cairo, Egypt

<sup>3,4</sup>Mechatronics Engineering Department, Ain Shams University, Cairo, Egypt

Received: 4-11-2014/ Revised: 8-12-2014 / Accepted: 28-12-2014

---

### Abstract

In this paper, visual servoing control scheme for mobile robot navigation is discussed. The survey introduces a brief discussion of the subjects of the system under consideration. These subjects include mobile robot platform kinematics, vision systems, visual servoing, mapping, navigation, automatic control, and finally the avoidance techniques used to avoid obstacle during the navigation process. Different vision based navigation systems will be discussed and a comparative study will be given.

**Key Words:** Mobile robot, visual servoing, navigation, obstacle avoidance and Redundancy.

---

### Introduction

Visual control is one of the most important approaches that for navigation purposes. It is also the most powerful and popular sensing method used in the navigation of autonomous robot where many significant contributions have been presented in the last decade<sup>(1)</sup>. Visual servoing refers to using computer vision data in the servo loop to control the motion of a robot. Closed control schemes using vision signal which introduced in order to increase the flexibility and the accuracy of robotic systems. It is now a widely used technique in robot control. More generally, the aim of the visual servoing approach is to control a robot using the information provided by a vision system<sup>(2)</sup>. For more reading about visual servoing in general, two interesting surveys are<sup>(3,4)</sup>.

The estimation of configuration, navigation, visual servoing and path planning is the key element for controlling an autonomous mobile robot successfully.

Robot navigation means robot's ability to determine its own position in its reference frame and then to draw a path between starting and a goal point for the robot moving between them.

Navigation relies basically on three major approaches self-localization, path planning, map-building and map interpretation. Robot localization means matching the landmarks found in an on-line image with expected landmarks of database and is essential for a correct navigation for the mobile robot in the environment.

Then, the Path planning is effectively an extension of localization, in which it requires the determination of

the robot's current position and a position of goal location, both within the same frame of reference or coordinates. So, the main idea for map-based navigation is essential to provide the robot with a sequence of landmarks expected to be found during navigation, and the task of the vision system is then to search for and recognize the observed landmarks<sup>(5)</sup>.

Environment and mapping are two important factors for navigation. Firstly, the environment is classified into two basic subjects Location and structure; Location may be indoor or outdoor as well as structure is subdivided into structured and unstructured. Secondly, mapping which includes two main approaches: map-based navigation and mapless navigation. The map-based navigation is subdivided into firstly, metric map-using navigation systems which requires a complete map of the environment before the navigation starts. Secondly, metric map-building navigation systems, which builds the map by themselves, then use it during the navigation. Finally, topological map-based systems build and/or use topological maps which consist of nodes linked by lines where each node represents the most characteristic places of the environment while, The mapless navigation mostly is classified into four methods optical flow, appearance based, feature tracking and object recognition.

Visual servoing known as Vision-Based Robot Control is a technique, which uses feedback information extracted from a vision sensor to control the motion of a robot. S. A. Hutchinson, G. D. Hager, and P. I. Corke

from SRI International Labs published the first tutorial on Visual Servoing in 1996 and more recent tutorials were published in 2006 and 2007 by F. Chaumette and S. Hutchinson<sup>(2)</sup>. Visual Servoing techniques are broadly classified into the following types

- Image Based (IBVS)
- Position Based (PBVS)
- Hybrid Approach

The vision data may be excluded from a camera that is mounted directly on a robot end effector, on a mobile robot, or the camera can be fixed in the workspace so that it can observe the robot motion from a stationary configuration. Other configurations can be considered for instance, several cameras mounted on pan-tilt heads observing the robot motion.

The rest of the paper is organized as follows: first, section 3, visual servoing is presented and their methods are discussed. Second, section 4, includes a task and control design of different vision-based methods. ; Third, section5, comparison between different vision-based navigation System. Finally, section 6, concludes the paper.

### Visual servoing

There are three type of visual servoing. The first type is the image based visual servoing (IBVS) where the feature parameters vector extracted is compared with the desired feature parameter vector to determine the feature space error vector. This error vector is used by feature space control law to generate a Cartesian or joint-level control command. The second one is the position based visual servoing (PBVS). In this type the features is extracted from the images and used in conjunction with the geometric model of the object to estimate the pose of the object with respect to the camera. Feedback is computed by reducing in the estimated pose space. The third type is hybrid visual servoing where the pose estimation is partial and determines rotation parameters only. The control input is expressed partially in 3D Cartesian space and in part in 2D image space. An image-based control is used to control the camera translations while the orientation vector is extracted and used to control the camera rotational degrees of freedom.

#### 3.1. Image based visual servoing

Classically, to achieve a visual servoing task, a set of visual features has to be selected from the image in order to control the desired degrees of freedom (DOF). A control law has also to be designed so that these visual features  $s$  reach a desired values  $s^*$ ,

leading to a correct realization of the task. The control principle is thus to regulate zero the error vector  $e = s - s^*$ . To build this control law, the

interaction matrix  $L_s$  is required. For eye-in-hand

systems, this matrix links the time variation of  $s$  to the camera instantaneous velocity  $v$ ,

$$S' = L_s V_s \quad (3.1)$$

With  $V_c = (v \ w)$  where  $v$  is the linear camera velocity and  $\omega$  its angular velocity. Thereafter, if

we consider the camera velocity as input of the robot controller, the following control law is designed to try to obtain an exponential decoupled decrease of the error  $e$ .

$$V_c = -\lambda \hat{L}_s^+ e \quad (3.2)$$

Where  $\lambda$  is a proportional gain that has to be tuned to minimize the time-to convergence, and  $\hat{L}_s^+$  is the pseudo-inverse of a model.

As it can be seen, visual servoing explicitly relies on the choice of the visual features  $s$  (and then on the related interaction matrix. This is the basic design implemented by most visual servo controllers.

#### The interaction matrix

More precisely, for a 3-D point with coordinates  $X = (X, Y, Z)$  in the camera frame, which projects in

the image as a 2-D point with coordinates  $X = (x, y)$ ,

$$\text{we have: } \begin{cases} x = X/Z = (u - c_u)/f\alpha \\ y = Y/Z = (u - c_v)/f \end{cases}$$

Where  $m = (u, v)$  gives the coordinates of the image point expressed in pixel units, and  $a = (c_u, c_v, f, \alpha)$  is

the set of camera intrinsic parameters:  $c_u$  and  $c_v$  are

the coordinates of the principal point,  $f$  is the focal

length, and  $\alpha$  is the ratio of the pixel dimensions.. In

this case, we take  $s = X = (x, y)$ , the image plane

coordinates of the point. Taking the time derivative and grouping terms we can obtain:

$$\dot{X} = L_x V_c \quad (3.3)$$

Where the interaction matrix  $L_x$  related to  $x$  is

$$L_x = \begin{bmatrix} \frac{-1}{z} & 0 & \frac{x}{z} & xy & -(1+x^2) & y \\ 0 & \frac{-1}{z} & \frac{y}{z} & 1+y^2 & -xy & -x \end{bmatrix} \quad (3.4)$$

In the matrix  $L_x$ , the value  $Z$  is the depth of the point relative to the camera frame. The camera intrinsic parameters are involved in the computation of  $x$  and  $y$ . Therefore, both for the any control scheme

that uses this form of the interaction matrix must estimate or approximate the value of  $Z$  and  $(x, y)$ .

**Position based visual servoing**

Basically, to compute the pose, a set of measurements in one image necessitates the camera intrinsic parameters and the 3-D model of the object observed to be known. It is convenient to consider three coordinate frames: the current camera frame  $F_c$ , the desired camera frame  $F_{c^*}$ , and a reference

frame  $F_o$  attached to the object. Thus, the coordinate vector  $t_o^c$  and  $t_o^{c^*}$  give the coordinates of the origin of

the object frame expressed relative to the current camera frame and relative to the desired camera frame, respectively.

We can define  $s$  to be  $(t, \theta u)$  in which  $t$  is a translation vector, and  $\theta u$  gives the angle  $\theta$  axis parameterization for the rotation. We now discuss two choices for  $t$  and give the corresponding control laws.

If  $t$  is defined relative to the object frame  $F_o$ , we obtain  $s = (t_o^{c^*}, \theta u)$ . In that case, we

have  $s^* = 0, e = s$ , and the interaction matrix related to  $e$  is given

$$L_e = \begin{bmatrix} R & 0 \\ 0 & L_{\theta u} \end{bmatrix} \quad (3.5)$$

Where  $R = R_c^{c^*}$  be the rotation matrix that gives the orientation of the current camera frame relative to the desired frame. in which  $I_3$  is the  $3 \times 3$  identity

matrix and  $L_{\theta u}$  is

$$L_{\theta u} = I_3 - \frac{\theta}{2} [u]_x - \left( 1 - \frac{\text{sinc } \theta}{\text{sinc}^2 \frac{\theta}{2}} \right) [u]_x^2$$

Where  $\text{sinc } x$  is the sinus cardinal defined such that

$x \text{ sinc } x = \sin x$  and  $\text{sinc } 0 = 1$ . The decoupling

between translational and rotational motions, which allows us to obtain a simple control scheme:

$$\begin{cases} v_c = -\lambda R^T t_o^{c^*} \\ \omega_c = -\lambda \theta u \end{cases} \quad (3.6)$$

**Hybrid based visual servoing**

Sunhyo Kim and Se-Young Oh, 2007 [30] presented a new hybrid fuzzy control method for visual servoing of mobile robots, IBVS calculates the motion plan directly from the image space using the inverse image Jacobian so that the target object always stays within the field of view. Then combines, PBVS that uses an image-to-work space transform to plan an optimal pose trajectory directly in the Cartesian.

One of the disadvantages of PBVS is that it cannot guarantee that the target object stays in the field of view of the camera during the servoing task, which was solved by using IBVS.

Hybrid method was used by Andrea Cherubini, François Chaumette and Giuseppe Oriolo, 2011 [22] in a different way. Their approach used the pose-based controller that can be used in general initial conditions and was appropriate for path reaching whereas the image-based controller, is more precise, robust, and should be preferred for path following (i.e., when the error is small).

**Homography and its estimation**

The homography-based visual servoing method differs from IBVS and PBVS in its ability to achieve the visual control task without needing of depth estimation or any measurements of the scene.

The homography is described in a two dimensional point  $X_{2D} = (x, y)$  which lies on a plane can be

represented by a three dimensional vector as well like. Here,  $X_{2D}$  is the scaled version of  $X_{3D}$  by its

third elements such as  $x = x_1/x_3$  and  $y = x_2/x_3$ .

This kind of representation  $(X_{2D})$  is used in

homography analysis and it is called homogeneous representation of a point lying on a projective plane  $p^2$ . Then, homography can be defined as a

mapping of these points from one projective plane to another projective plane and it has the property of

invertibility. A homography is an invertible mapping from  $p^2$  to itself, such that three points lie

on the same line if and only if their mapped points are also collinear and its algebraic definition is as such: A mapping from  $p^2 \rightarrow p^2$  is a projectivity if

and only if there exists a nonsingular 3x3 matrix  $H$  such that for any point in  $p^2$  represented by a

vector  $x$ , it is true that its mapped point is equal to  $Hx$ .

There are several geometric transformations each of which has some properties peculiar to them and homographies are one of them which are Isometries, Similarity Transformations, Affine Transformations, Perspective Projection and Projective Transformation [5]. Now we focus on the most important geometric transformation method that widely used. This geometric transformation is Perspective Projection model.

Perspective projection is the projection of three dimensional points in the Cartesian space to two dimensional points. Homography can link two perspective images geometrically relating points in one image belonging to a plane of the scene to the corresponding points in the other image Fig.3.1. The homography between two images can be computed using image point matches.

Let us suppose the two images are obtained with the same camera, whose projection matrixes in a common reference system are  $p_1 = K[I|0]$  and

$p_2 = K[R|t]$ , with,  $t = -Rc$  being  $R$  the camera rotation,  $c$  the distance between the cameras, and  $K$

the internal calibration matrix,

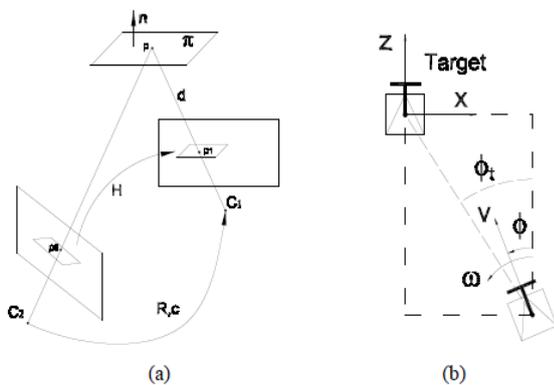


Fig.3.1: (a) Homography from a plane between two views, where  $C_1$  and  $C_2$  are the optical centers of the cameras. (b) Coordinate system.

A point can be represented in the image with homogeneous coordinates as  $p = (x, y, 1)^T$ .

A projective transformation  $H$  relates pairs of matched points belonging to a plane of the scene ( $\pi$ ) in such a way that  $p_1 = Hp_2$ .

A homography  $H$  can be related to camera motion in such a way that

$$H = K \left( R - t \frac{n^T}{d} \right) K^{-1} = KR \left( I + c \frac{n^T}{d} \right) K^{-1}$$

(3.7)

$$n = (n_x, n_y, n_z)^T$$

In where  $n$  is the normal to the scene plane that generates the homography and  $d$  the distance between the plane and the origin are referred to the global reference attached to the target position.

- Input-Output Linearization

This method achieves linearization of a non-linear system by differentiating the homography elements until solving for the control inputs [15].

S. Benhimane and E. Malis, 2007 [11] proposed a new homography-based visual control approach that deals with the nonholonomic constraints of the platform. The visual tracking algorithm proposed in this approach was based on a new efficient second-order minimization (the ESM algorithm) method that has a higher convergence rate than standard first-order minimization techniques. They used homography-based 2D visual servoing that do not need any measure of the 3D structure of the observed target proposed.

- Visual servoing with Shortest path control

The aim of the shortest path control approach is to design a navigation scheme in which the robot can correct rotation and translation in a decoupled way, so it proposes decoupling the motion, rotation and translation by following a specific trajectory. The resulting path of this motion is shown in Fig. 3.2.

A shortest path control was presented by G. López-Nicols, C. Sagüés and J.J. Guerrero, 2007 [17] the control law was designed directly based on the homography elements. This system has two variables to be controlled (the velocities  $v$  and  $w$ ), It needs at least two parameters of the homography to define the control law.

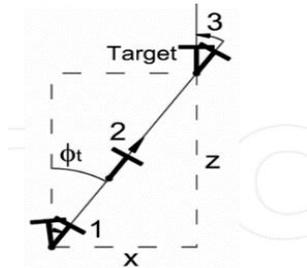


Fig.3.2: Motion trajectory of the robot consisting in three steps

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 was reached up to a rotation. Finally, the orientation was corrected in the third step.

**Task control and design**

The autonomous navigation of wheeled mobile robots relies on dead reckoning, which is based on internal sensors. However, dead reckoning suffers from errors as a result of wheel slippage and error accumulation due to inaccurate modelling of the robot kinematics as well as the robot's environment. In contrast, visual servoing can overcome this problem by eliminating the need for absolute positioning of both the robot itself and the goal. As visual sensing provides continual updating for the robot's relative position to the goal.

Aveek K. D, 2001, presented vision-based control of a nonholonomic (car-like) mobile robot [7]. Then, described the design and implementation of real-time estimation and control algorithms on a platform using a single omni-directional camera as a sensor without explicit odometry. One of the advantages of this approach is using an omni-directional camera as a visual sensor, which has a wider field of view and allows extracting more information from the video signal, hence making it possible to implement a wider range of motion strategies on the same platform. The second advantage is that, the velocity estimation & localization techniques based on extended Kalman filter (EKF) integrated in the closed loop system.

In this approach showed two types of motion control tasks: the first one is wall following mode where the goal is to guide the robot so that it maintains a specified orientation and distance from a boundary. The second type is follow the leader task, where the robot is instructed to maintain a prescribed position and orientation with respect to a moving target.

**Wall follower**

The wall follower operation is done by taking inputs from 2 "sensors" - a wall detector and an obstacle detector. Since the position and orientation relative to the wall are known, the detector is able to distinguish between the features, which are actually the wall, and obstacles that must be avoided. I/O feedback

linearization techniques are used to design a PD controller to regulate the distance of the vehicle to the wall.

Wall following can be considered as a particular case of path following. As shown in fig .4.1.

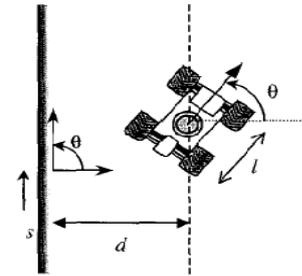


Fig. 4.1: Wall following

Thus, the kinematics in terms of the path variables become

Thus, the kinematics in terms of the path variables become

$$\dot{s} = u_1 \cos \theta_p \quad , \quad \dot{d} = u_1 \sin \theta_p$$

$$\dot{\theta}_p = u_1 \frac{\tan \phi}{l} \quad , \quad \dot{\phi} = u_2 \quad (4.1)$$

In fact, in this situation the path may be taken as the x-axis of the world frame, and (s, d, theta\_d) become

the coordinates (x, y, theta) of the vehicle. where x, y are the cartesian coordinates of the rear wheel, theta

measures the orientation of the car body with respect to the x- axis, and phi is the steering angle.

$$\theta_t = \frac{\pi}{2} \theta_p = \theta - \theta_t \quad (4.2)$$

**Leader follower**

The Leader Follower object described next, requires reliable estimation of the linear velocity v\_i(t) and angular velocity omega\_i(t) of the leader mobile robot R\_i, and relative orientation (theta\_i - theta\_j).

The velocity estimator algorithm is based on an extended Kalman filter. It uses the omni-directional vision system to determine the range rho\_ij and the

bearing beta\_ij of the observed leader R\_i, see Fig.4.2. In

addition, the filter requires a sensor model and the relative kinematic equations of the leader R\_i and

follower  $R_j$ . The image processing, algorithms

provide the following observations

Range:  $\rho_{ij}^2 = (x_i - x_j)^2 + (y_i - y_j)^2$

Bearing:

$$\beta_{ij} = \frac{\pi}{2} + \text{atan2}(y_i - y_j, x_i - x_j) - \theta_j$$

The closed-loop omni-directional vision system for leader tracking is depicted in Fig. 4.3. Let us define

$$x_{ij} = x_i - x_j$$

$$y_{ij} = y_i - y_j \quad (4.3)$$

$$\alpha_{ij} = \beta_{ij} + \theta_j - \theta_i$$

Then, 
$$\dot{\rho}_{ij} = \frac{x_{ij}x_{ij} + y_{ij}y_{ij}}{\rho_{ij}}$$

And 
$$\dot{\beta}_{ij} = \frac{y_{ij}x_{ij} - y_{ij}x_{ij}}{\rho_{ij}^2} - \dot{\theta}_j$$

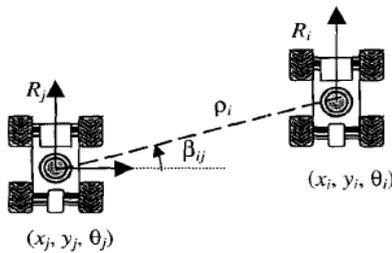


Fig.4.2: Robot configuration for velocity estimation.

The Velocity Estimator object provides the follower with necessary information about the velocity of the leader for feed forward control. This eliminates the need for explicit communication. The basic structure for this object is shown in Fig.4.3.

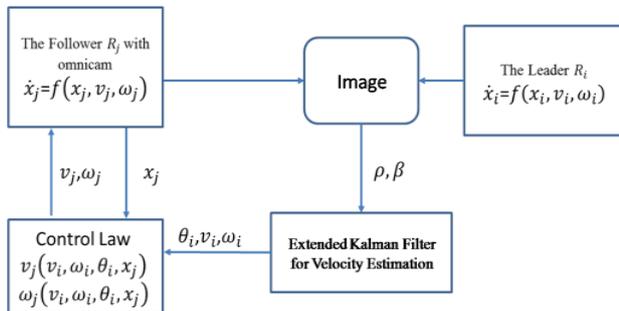


Fig.4.3: Leader follower structure

**Image based control**

Adrian Burlacu, 2011 [28] developed a real-time Image-based visual servoing strategy for a nonholonomic two-wheel mobile robot. In this

method, the non-holonomic planar motion constraint of mobile robots taken into consideration without requiring the estimation and decomposition of the homography or fundamental matrix. A switched image based controller was design, using point features, in order to solve the camera field-of-view (FOV) constraint. This approach has the advantage of not requiring the estimation and decomposition of the homography or fundamental matrix. The drawback is failing to adapt with non-static objects. Different types of visual features could generated from image processing algorithms. In visual analysis applications, point features are low-level descriptors precisely locatable and persistent. These basic properties define the usefulness of point features for object description.

Wenhao Fu, 2013 [25] combined the reactive collision avoidance methods with image-based visual servoing control for two-wheel differential drive mobile robot navigation in an indoor environment. This approach used a perspective camera fixed on a mobile robot to navigate the robot towards a certain reference image key with the ability to avoid collision in an unknown environment. However, exploiting the odometry based pose estimation obstacle avoidance may cause visual target loss. The laser range finder detects obstacles and their boundaries are modeled using B-spline curves. The main advantage is that it can have arbitrary order of continuity, which is suitable for mobile robot control.

David Folio and Viviane Cadenat, 2005 [13, 14] applied sensor-based controller approach on a cart-like mobile robot to safely perform a vision-based task in a cluttered environment. The method relies on the switch between different controllers depending on the risk of collision and occlusion. The continuous switch between three controllers realizing respectively the nominal vision based task, the obstacle bypassing and the occlusion avoidance. However, this work is restricted to missions where occlusions can be effectively avoided. The main advantage of this approach is that partial or total short duration occlusions are really tolerated in the obstacle neighborhood.

**Path follower**

Andrea Cherubini and Francois Chaumette, 2008 [27] presented an image-based visual servoing controller enabling nonholonomic (car-like) mobile robots with a fixed pinhole camera to reach and follow a continuous path on the ground. The controller does not use the complete geometric representation of the path but only utilizes the features extracted from the image plane while, the stability was achieved using Lyapunov-based stability analysis. However, One of drawback using only two features which is greater noise sensitivity.

Image –based scheme was also used in 2011 [22], and the path to be followed was a curve that is drawn on the ground. The curve can be differentiable only once. However, these techniques are sensitive to feature detection and can not be applied to complex images that do not contain special sets of features to track.

**Control design**

Such a path following constraint, the goal of driving  $q$  to  $q^*$  be equivalent to zeroing the error:  $e = q - q^* \in R^3$ . by using appropriate control inputs  $v$  and  $\omega$ . The state dynamics is

$$\dot{s} = J(s)u = J_v(s)v + J_\omega(s)\omega$$

Where  $J_v$  and  $J_\omega$  are the columns of the Jacobian  $J(s)$  that relates  $u$  to  $\dot{s}$ . By apply the following feedback control:

$$\omega = -J_\omega^+(\lambda e + J_v v^*) \quad (4.4)$$

with  $\lambda > 0$ , and  $J_\omega^+$  is the Moore–Penrose matrix pseudoinverse of  $J_\omega$ .

- **Image-Based Path Follower**

The image based path follower utilizes a row and a column primitive controller. However, in this case, the controllers are based on the reference path point image features instead of its 3D features.

To develop the row and column controller, the relations between the visual features and the robot velocity should be found firstly.

- **Row controller**

The task of the row controller is to drive  $(X, \theta)$  to a desired state  $(X^*, \theta^*)$ , while constraining  $R^*$  to a row in the image.

This is equivalent to zeroing  $e = [X - X^* \theta - \theta^*]^T$ ,

under the path following constraint:

$$Y = const = Y^*$$

where the expressions of  $J_v$  and  $J_\omega$  can be written

$$J_v = \begin{bmatrix} L_X \\ L_\theta \end{bmatrix} T_v \quad , \quad J_\omega = \begin{bmatrix} L_X \\ L_\theta \end{bmatrix} T_\omega \quad (4.5)$$

By imposing the motion, the system state equations become

$$\begin{bmatrix} \dot{X} \\ \dot{\theta} \end{bmatrix} = J_v v^* + J_\omega \omega + \begin{bmatrix} \frac{\partial X}{\partial t} \\ \frac{\partial \theta}{\partial t} \end{bmatrix} \quad (4.6)$$

Note that, as aforementioned, this control law does not compensate the terms  $\frac{\partial X}{\partial t}$  and  $\frac{\partial \theta}{\partial t}$ .

- **Column controller**

The task of the column controller is to drive  $(Y, \theta)$  to a desired state  $(Y^*, \theta^*)$  while constraining  $R^*$  to a column in the image. This is equivalent to zeroing  $e = [Y - Y^* \theta - \theta^*]^T$ , under the path following constraint:

$$X = const = X^*$$

Where the expressions of  $J_v$  and  $J_\omega$  can be written

$$J_v = \begin{bmatrix} L_Y \\ L_\theta \end{bmatrix} T_v J_\omega = \begin{bmatrix} L_Y \\ L_\theta \end{bmatrix} T_\omega \quad (4.7)$$

By imposing the motion, the system state equations become

$$\begin{bmatrix} \dot{Y} \\ \dot{\theta} \end{bmatrix} = J_v v^* + J_\omega \omega + \begin{bmatrix} \frac{\partial Y}{\partial t} \\ \frac{\partial \theta}{\partial t} \end{bmatrix} \quad (4.8)$$

Note that, as aforementioned, this control law does not compensate the terms  $\frac{\partial Y}{\partial t}$  and  $\frac{\partial \theta}{\partial t}$ .

- **Pose-based path follower**

For the pose-based approach, the path 3D features in the camera frame must be derived from the image features, by considering a pinhole camera model. The four camera parameters used for projecting are the focal length in pixels  $f$ , and  $\rho$ ,  $t_y$ , and  $t_z$ . For

simplicity, let consider a nor For the pose-based approach, the path 3D features in the camera frame must be derived from the image features, by considering a pinhole camera model. The four camera parameters used for projecting are the focal length in pixels  $f$ , and  $\rho$ ,  $t_y$ , and  $t_z$ .

The mapping between the image frame and camera frame coordinates of a ground point gives

$$x_c = \frac{x r_y}{\sin \rho + Y \cos \rho}, y_c = \frac{y r_y}{\sin \rho + Y \cos \rho},$$

$$z_c = \frac{t_y}{\sin \rho + Y \cos \rho}$$

• **Row controller:**

The goal of the pose-based row controller is to drive  $(x, z, \theta)$  to a desired state  $(x^*, z^*, \theta^*)$

While constraining  $R^*$  to a row in the image  $Y = const = Y^*$ . This is equivalent to zeroing the error  $e = [X - X^* \ \theta - \theta^*]^T$ , which is equivalent to applying the path following constraint:

$$z = const = z^* = \frac{t_y}{\sin \rho + Y^* \cos \rho}$$

Hence, the system state equations are

$$\begin{bmatrix} \dot{x} \\ \dot{\theta} \end{bmatrix} = J_v v + J_\omega \omega \text{ with: } J_v = \begin{bmatrix} \tan \theta \\ -\frac{c^*}{\cos \theta} \end{bmatrix}$$

$$J_\omega = \begin{bmatrix} z^* + x \tan \theta \\ 1 - \frac{c^* x}{\cos \theta} \end{bmatrix}. \quad (4.9)$$

(thus, the value of  $c^*$ ) is derived by least square interpolation. Under the constraint that  $|\theta| \neq \frac{\pi}{2}$ , which can be avoided by temporarily using the pose-based column controller while  $\Gamma$  is parallel to X.

By imposing the motion, the system state equations become

$$\begin{bmatrix} \dot{x} \\ \dot{\theta} \end{bmatrix} = J_v v^* + J_\omega \omega \quad (4.10)$$

This system can be controlled by (4.4) using the feedback law

**Column controller:**

The goal of the pose-based column controller is to drive  $(x, z, \theta)$  to a desired state  $(x^*, z^*, \theta^*)$  while constraining  $R^*$  to a column in the image  $X = const = X^*$ .

Hence, the system state equations are

$$\begin{bmatrix} \dot{x} \\ \dot{\theta} \end{bmatrix} = J_v v + J_\omega \omega$$

With :  $J_v = \begin{bmatrix} \tan \theta \cos \beta - \sin \beta \\ -\frac{c^* \cos \beta}{\cos \theta} \end{bmatrix}$

$$J_\omega = \begin{bmatrix} z^* + x \tan \theta \\ 1 - \frac{c^* x}{\cos \theta} \end{bmatrix} \quad (4.11)$$

under the constraint that  $|\theta| \neq \pi/2$ , which can be avoided by temporarily using the pose-based column controller while  $\Gamma$  is parallel to X. By imposing the motion, the system state equations become

$$\begin{bmatrix} \dot{x} \\ \dot{\theta} \end{bmatrix} = J_v v^* + J_\omega \omega \quad (4.12)$$

**Obstacle modeling**

Andrea Cherubini and Francois Chaumette, 2014 [26] added and validated a framework for avoiding moving obstacles during visual navigation using a wheeled mobile robot. Visual navigation achieved by following a path, represented as an ordered set of key images, which have been acquired by an on-board camera in a teaching phase. To estimate the obstacle velocities, we have designed a Kalman-based observer. Then, we utilize the velocities to predict possible collisions between robot and obstacles.

• **Redundancy**

Most of the recent approaches used redundancy as a basic technique. A task redundant robot possesses more DOFs than those required to execute its primary task (visual servoing). This provides increased dexterity that may be used for a secondary task (here, obstacle avoidance). The tasks can be executed by using a projecting operator or a task priority strategy. Sensor based tasks have been tackled with redundancy in various works (e.g., on a robot arm, and on a nonholonomic manipulator) [19].

• **Redundancy framework for 2 tasks:**

Redundancy allows robot systems to increase their level of manipulability and flexibility. There are several types of redundancy for a mobile robot. Redundancy with respect to the kinematic when the robot has more DOFs than those required to place its self at a given position and orientation within the task space, and redundancy with respect to the task when the number of independent parameters required by a task is less than the DOFs of the robot movement.

**Comparison study**

There are many papers presented systems with mobile robots using vision systems in navigation processes and mapping in different environments. In this section we will discuss an important group of these techniques that discussed this subject with different perspective. Regarding the type of robot (platform), vision system, navigation and obstacle avoidance as mentioned and discussed before, shown in table 1.

**Conclusion**

This research aims to offer a classification for different visual servoing control schemes for mobile robot navigation according to the navigation methods and the techniques used for vision based navigation. The commonly used visual servoing control schemes includes position based visual servoing, image based visual servoing, and visual servoing using homography. These techniques are summarized with a discussion of their mathematical model. This paper includes comparison between different vision-based navigation systems.

**Table 1.**

Authors	Type of robot (platform)	Visual scheme	Type of feature	Environment	Vision & other sensor systems	Obstacle avoidance technique
[8]	Uni-cycle & car-like robot	IBVS	Following curve path	In-door	Monocular camera	
[9]	Car-like robot	IBVS	Point (landmark)	In-door	Range-image camera & CCD sensor	Laser range scanner system
[10],[11],[1]	Hilar mobile robot	Homography	Point	Cluttered	Monocular camera	
[13],[14]	Cart-like robot	IBVS	Non-linear following path	Cluttered	Single camera & Proximetric sensor	Potential field
[15],[17]	Differential drive robot	Homography	Following line path	Cluttered	Monocular camera	
[18]	Car-like robot	IBVS	Following path	In-door	Central camera & Omni-directional camera	
[19]	Uni-cycle robot	IBVS	Circular path	In-door	Range scanner camera	Redundancy
[20,23,24,26]	Car-like robot	IBVS	Circular path	Out-door	Single camera	New tentacle method
[22]	Car-like robot	IBVS, PBVS	Following curve path	Out-door	Single camera	
[25]	Cart-like robot	IBVS	Corner points	In-door	Single camera & 2D-laser range, odometry sensor	B-spline
[7]	Car-like robot	IBVS	Wall follower, Leader follower and landmarks	In-door	Single Omni-directional camera	
[27]	Car-like robot	IBVS	Following path	Out-door	Fixed single camera	
[28]	Two-wheel robot	IBVS	Point	In-door	Monocular camera	
[6]	Tri-cycle robot	IBVS based localization	Image	In-door	Monocular camera & Ultrasonic sensor	Stationary Sonar System

Authors	Type of robot (platform)	Visual scheme	Type of feature	Environment	Vision & other sensor systems	Obstacle avoidance technique
[21]	Cart-like robot	IBVS or PBVS	Point (Landmark)	Cluttered In-door	Single camera & Laser sensor	Potential field
[30]	Cart-like robot	Hybird visual servo	Artificial Landmark	In-door	Single camera & CCD sensor	
[31]	Car-like robot	PBVS	the position of a path point, and the path tangent orientation and curvature at that point	Out-door	Fixed camera	

## References

1. Yongchun Fang, Warren E. Dixon, Darren M. Dawson and Prakash Chawda, "Homography-Based Visual Servo Regulation of Mobile Robots", IEEE Transection on System, 2005, 35: 5.
2. F. Chaumette and S. Hutchinson, "Visual servo control, part i: Basic approaches," IEEE Robotics and Automation Magazine, 2007, 1: 82–90.
3. Ezio, INRIA and Sophia Antipolis, "Survey of vision-based robot control", IEEE Robotics & Automation Magazine, (2002).
4. S. Hutchinson, G. D. Hager, and P. I. Corke, "A tutorial on visual servo control", IEEE Tran. On Robotics and Automation, 1996, 5: 651–670.
5. Tufa TEPE, DraganKostic and HenkNijmeijer, (2010). "Mobile Robot Navigation Using Visual Servoing ", IEEE International Conference on Robotics and Automation, pp. 3-13.
6. A. Ohya, A. Kosaka and A. Kak, "Vision-based navigation by a mobile robot with obstacle avoidance using a single-camera vision and ultrasonic sensing", IEEE Trans. on Robotics and Automation, 1998, 6 : 969–978.
7. Avek K. Das, Rafael Fierro, R. Vijay Kumar, B. Southall and John R. Spletzer, "Real-Time Vision-Based Control of a Nonholonomic Mobile Robot", IEEE International Conference on Robotics and Automation, 2001, 2 : 1714-1719.
8. Yi Ma, Jana Ko'seck'a and Shankar S. Sastry, "Vision Guided Navigation for a Nonholonomic Mobile Robot", IEEE Transactions on Robotics and Automation, 1999, 15: 3.
9. Thorsten Lietmann, Burghard Bornstedt and Boris Lohmann. (2001) "Visual Servoing for a Non Holonomic Mobile Robot Using A Range-Image Camera", Proceedings of European Control Conference.
10. J. Chen, W. E. Dixon, D. M. Dawson, and M. McIntire, "Homography-Based Visual Servo Tracking Control of a Wheeled Mobile Robot", IEEE International Conference on Intelligent Robots and Systems, 2003, 27 - 31.
11. S. Benhimane and E. Malis , " Homography-based 2D Visual Tracking and Servoing", The International Journal of Robotics Research, 2007, 26: 661–676.
12. Francisco Bonin-Font, Alberto Ortiz and Gabriel Oliver, " Visual Navigation for Mobile Robots: a Survey", Journal of Intelligent and Robotic Systems, 2008, 3: 263–296.
13. David Folio and Viviane Cadenat, "A controller to avoid both occlusions and obstacles during a vision-based navigation task in a cluttered environment", 44th IEEE Conference on Decision, Control, and the European Control Conference, 2005, 11-12.
14. David Folio and Viviane Cadenat . (2006). "A new controller to perform safe vision-based navigation tasks amidst possibly occluding obstacles", Robotics and Autonomous Systems.
15. G. L'opez-Nicol'as, C. Sag'u'es and J.J. Guerrero, " Homography-Based Visual Control of Nonholonomic Vehicles", IEEE Int. Conference on Robotics and Automation, 2007, 1703-1708.
16. B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics," IEEE Trans. Robot. Automation, (June 1992).
17. G. López-Nicols, C. Sagüés and J.J. Guerrero , " Shortest Path Homography-Based Visual Control for Differential Drive Robots", Vision Systems: Applications, pp. 608, Vienna, Austria, (June 2007).
18. Jonathan Courbon, YoucefMezouar, and Philippe Martinet, " Indoor navigation of a non-Holonomic mobile robot using a visual memory", Springer Science Business Media, Autonomous Robot, 2008, 253–266.
19. A. Cherubini and F. Chaumette, "A redundancy-based approach for obstacle avoidance in mobile robot navigation", IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, (2010).
20. Cherubini A and Chaumette F , "Visual navigation with obstacle avoidance", In IEEE/RSJ International Conference on Intelligent Robots and Systems, San Francisco, (2011).
21. A. Durand Petiteville, S. Hutchinson, V. Cadenat and M. Courdesses, " 2D visual servoing for a long range navigation in a cluttered environment", 50th IEEE Conference on Decision and Control and European Control Conference, 2011, 12-15.
22. Andrea Cherubini, François Chaumette and Giuseppe Oriolo, " Visual servoing for path reaching with nonholonomic robots", Robotica, 2011, 29, 1037–104.
23. Cherubini A, Spindler F and Chaumette F, "A new tentacles based technique for avoiding obstacles during visual navigation", In IEEE International Conference on Robotics and Automation, St. Paul,( 2012).
24. Andrea Cherubini, and François Chaumette, "Visual navigation of a mobile robot with laser-based collision avoidance", The International Journal of Robotics Research, 2012.
25. Wenhao Fu, HichamHadj-Abdelkader and Etienne Colle, " Visual servoing based mobile robot navigation able to deal with complete target loss", 18th International Conference on Methods and Models in Automation and Robotics, (19 July 2013).
26. Andrea Cherubini, Fabien Spindler and Francois Chaumette, " Autonomous Visual Navigation and Laser-based Moving Obstacle Avoidance", IEEE Transactions on Intelligent Transportation Systems, (2014).
27. Andrea Cherubini, Francois Chaumette and Giuseppe Oriolo, " An Image-based Visual Servoing Scheme for Following Paths with Nonholonomic Mobile Robots", 10th Intl. Conf. on Control, Automation, Robotics and Vision Hanoi, Vietnam, 2008, 17–20.
28. Adrian Burlacu, ObadaMihai and Vladimir Tiganas, " An Image Based Visual Servoing Scheme for Wheeled Mobile Robots", IEEE International Conference on Robotics and Automation, 2011, 14-16.
29. De Souza, G. N. and Kak, A. C., "Vision for mobile robot navigation: A survey", IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002, 24 (2): 237–267.
30. Sunhyo Kim and Se-Young Oh, "Hybrid Position and Image Based Visual Servoing for mobile robots", Journal of Intelligent & Fuzzy Systems 73–82 IOS Press,( 2007).
31. A. Cherubini, F. Chaumette, and G. Oriolo, "A position-based visual servoing scheme for following paths with nonholonomic mobile robots," in IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS 2008, 1648–165.